

Data Mining for Phospho-Proteomics

By

Nila Reitz

A thesis submitted in partial fulfillment of
the requirements for the degree of

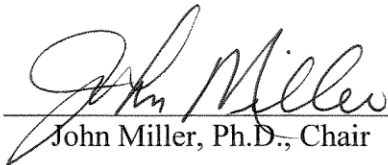
Master of Science in Computer Science

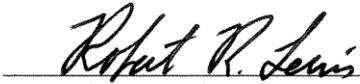
Washington State University
School of Electrical Engineering & Computer Science

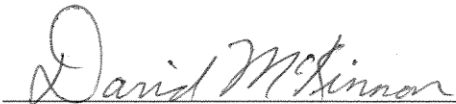
December 2009

To the faculty of Washington State University:

The members of the Committee appointed to examine the thesis of
Nila Allene Reitz find it satisfactory and recommend that it be accepted.


John Miller, Ph.D., Chair


Robert Lewis, Ph.D.


David McKinnon, Ph.D.

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor and chairman, Dr. John Miller for his guidance and advice through the whole process of my thesis research. I would also like to thank Dr. Robert Lewis and Dr. David McKinnon for their comments and assistance through the rigors of graduate school. I appreciate the participation of Dr. John Miller, Dr. Robert Lewis, and Dr. David McKinnon on my graduate committee.

I wish to thank the Environmental Molecular Sciences Laboratory at Pacific Northwest National Laboratory for the support of this research and for allowing me access to the proteomic experimental raw data. Pacific Northwest National Laboratory is operated by Battelle Memorial Institute for the U.S. Department of Energy through Contract DE-AC06-76RLO 1830.

Thanks also go to Sharon Johnson for proofreading the thesis and bringing consistency to my collection of ideas.

Finally my most heartfelt thanks go to my husband Devin Smith, for supporting, encouraging, and enduring the long process of graduate school.

Data Mining for Phospho-Proteomics

Abstract

by Nila Reitz, M.S.
Washington State University
December 2009

Chair: John Miller

The systematic investigation of phosphorylated proteins enabled by advances in mass spectrometry has the potential to reveal much about the signaling networks that regulate cellular function. Successful annotation of phospho-proteome data is an essential first step toward realizing this objective. Annotating the data requires the application of data mining techniques. This thesis reports on processes and tools developed for this purpose and applied to a dataset of phospho-peptides observed to be differentially abundant in irradiated tissue-culture samples.

TABLE OF CONTENTS

Acknowledgements.....	iii
Abstract.....	iv
List of Figures.....	vii
List of Tables.....	viii
Chapter 1 Introduction.....	1
Chapter 2 Related Work and Problem Statement.....	4
2.5 Background.....	4
2.6 Comparing Sequences.....	5
2.7 Data Mining.....	7
2.7.1 Data Mining Applications.....	7
2.7.2 Historical Data Mining for Proteomic Data.....	8
2.7.3 Data Mining Protein Data Produced from MS.....	8
2.7.4 Data Mining for Phospho-Peptide Data Produced from LC-MS/MS.....	11
2.8 Problem Statement /Issues that need to be addressed.....	12
Chapter 3 Data mining Phospho-proteomic data.....	14
3.1 Motivation.....	14
3.2 MS Process.....	15
3.3 The Data Set.....	15
3.4 Overview of Knowledge Discovery Process Branches.....	16
3.5 Identifying Statistically Significant Data – Combining Peptides.....	18
3.5.1 Combining Spectral Count Data.....	18
3.5.2 Statistical Significance.....	28

3.6 Signaling Pathways to Data	30
3.7 Data to Signaling Pathway	39
3.7.1 Submission to Phospho.ELM.....	40
3.7.2 Combining Phospho.ELM with NetworKIN in PPD database	43
3.7.3 Summarizing the Phospho.ELM + NetworKIN Output	44
Chapter 4 Chapter 4 Summary and Future Work	47
Bibliography	49
Appendix A Needleman-Wunsch Implementation.....	55
Appendix B Groups by Phosphorylation site match.....	70
Appendix C File Splitter Utility.....	71
Appendix D Statistically Significant Peptide Lists.....	74

LIST OF FIGURES

Figure 1.1 Overview of the steps that comprise the KDD process	2
Figure 3.1 Signaling Pathways to Data Method	17
Figure 3.2 Data to Signaling Pathways Method	17
Figure 3.3 PPD database query initial <i>Trivial</i> group definition.....	20
Figure 3.4 Directed acyclic graph of potential <i>Trivial</i> groups.....	22
Figure 3.5 Molecular Interaction Map (MIM) for Chk2.....	31
Figure 3.6 Example of NetworKIN output for kinase AKT1 and substrate ACN1	33
Figure 3.7 NetworKIN results forTP53BP1	39
Figure 3.8 BLAST Output for 2-Rad-GTest -1.html	43

LIST OF TABLES

Table 3.1 Example data for initial <i>Trivial</i> group definition.....	20
Table 3.2 Example data for revised <i>Trivial</i> group definition.....	21
Table 3.3 Example group with site match and substring.....	24
Table 3.4 Example group with site match, substring, and overlap.....	25
Table 3.5 Example group with site removed and exact match.....	28
Table 3.6 Successful Exact Match Comparison Result.....	37
Table 3.7 Top ten sequence alignments for query string TRHS*PT*PQQSNR.....	37
Table 3.8 Example PhosphoELM BLAST input file.....	42
Table 3.9 2RadPhosphoELM+Networkin Annotation Query Results.....	45
Table 3.10 5RadPhosphoELM+Networkin Annotation Query Results.....	46

CHAPTER 1 INTRODUCTION

Recently the introduction of new technologies and the improvements of old ones have allowed for an exponential increase in the volume of collected biological data. While manual or exhaustive search methods may work to analyze small collections of data, such methods are overwhelmed by the increase in data volume. Computers must be used to organize, maintain, and analyze the data. However computers, with their ability to rapidly complete repetitious tasks, may also be overwhelmed unless used wisely. In this case, wisely means the use of appropriate techniques and algorithms designed to efficiently find useful knowledge in the midst of massive amounts of data. Such appropriate techniques and algorithms form the field of data mining within computer science. To discover useful knowledge from vast collections of biological data, data mining techniques and algorithms must be applied.

Data mining is the discovery of useful knowledge from collections of data. It is the key step in the process known as Knowledge Discovery in Databases (KDD) (Figure 1.1) (Fayyad et al, 1996) [22]. An alternate definition of KDD is Knowledge Discovery and Data Mining. Both definitions are appropriate to my work. Data mining typically occurs as part of a larger process. The data supplied to data mining techniques must be collected into data sets (target data in Figure 1.1), suitable data sets must be selected, and the formats must be transformed and prepared for data mining tools. Other steps of the KDD process include observing patterns and evaluating the extracted knowledge from data mining. The process presented in this thesis will include the selection, preprocessing,

transformation and the data mining steps while visualization is left for future work and evaluating the extracted knowledge is the responsibility of the user.

The field of data mining utilizes techniques and algorithms from a wide variety of fields including statistics, machine learning, artificial intelligence, databases and data warehousing.

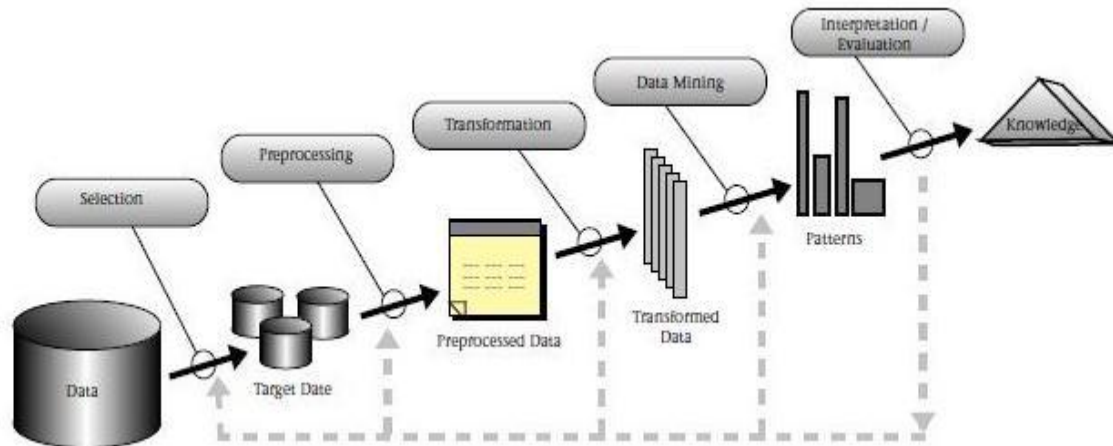


Figure 1.1 Overview of the steps that comprise the KDD process

The large-scale study of proteins is the area of biology known as proteomics. This thesis focuses on phospho-proteomics which is a branch of proteomics that identifies, catalogs, and characterizes proteins that contain a phosphate group as a posttranslational modification (PTM). A posttranslational modification is the chemical modification of a protein after its translation from RNA, the first stage in the process in which cells build proteins. A common form of protein posttranslational modification is reversible phosphorylation. This type of modification is catalyzed by protein kinases and phosphatases. Reversible phosphorylation regulates protein function, sub-cellular localization, complex formation, and degradation of proteins. As a result phospho-

proteomics is significant as it touches on protein features that regulate cell signaling networks.

A key issue in systems biology research is developing methods to analyze and understand the mechanisms by which cells process information. Such mechanisms critically depend on reversible phosphorylation of cellular proteins. Successfully applying methods of data mining to phospho-proteomics data will provide clues on what protein or pathway might be activated and indicate what proteins might be potential drug targets.

This thesis examines automated processes and tools that were developed for use in data mining phospho-proteomic data sets obtained by application of mass spectrometry technology Beausoleil, et al. [3], Olsen, et al. [4], Yang, et al.,[9]. These tools will assist in the identification of cellular signaling pathways for phospho-proteomic data.

The remainder of this thesis is arranged in the following manner. Chapter 2 presents existing data mining applications applied to proteomic data sets and identifies the problems that need to be solved. Chapter 3 presents the methods used in data mining for phospho-proteomics, and describes the procedures and tools developed as part of the research for this thesis for use in a process to data mine phospho-proteomic data sets. The process is illustrated by an application of the process to phospho-proteomic data obtained by Stenoien and coworkers [10]. This thesis will hereafter refer to this data set as PNNL data. Chapter 4 presents the conclusion of the research and discussion of future work.

CHAPTER 2 RELATED WORK AND PROBLEM STATEMENT

2.1 Background

A protein is a chain of amino acids folded into a globular form. Every protein is chemically defined by its unique sequence of amino-acid residues. These amino acid residues also define the three-dimensional structure of the protein. A protein sequence is made of up of shorter sequences of amino-acid residues known as peptide sequences. The twenty unique amino acids can be linked together in varying permutations to form a vast number of proteins. Proteins can act as enzymes to catalyze chemical reactions. The process of adding or removing a phosphate group (chemical formula PO_4) from a protein is known as phosphorylation. A kinase, or phosphotransferase, is a type of enzyme that transfers phosphate groups from high-energy donor molecules to specific substrates. An enzyme that removes phosphate groups is known as a phosphatase. A substrate is any molecule upon which an enzyme acts.

Enzymes are usually very specific as to which reactions they catalyze and the substrates that are involved in these reactions. Complementary shape, charge and hydrophilic/hydrophobic characteristics of enzymes and substrates are responsible for this specificity.

Phospho-proteomics is a branch of proteomics that identifies, catalogs, and characterizes proteins containing a phosphate group as a posttranslational modification. A posttranslational modification is the chemical modification of a protein after its translation from RNA, the first stage in the process in which cells build proteins.

A common form of protein posttranslational modification is reversible phosphorylation. This type of modification is catalyzed by protein kinases and plays a significant role in a wide range of cellular processes including regulating protein function, sub-cellular localization (which is the confining of a protein to a particular area within the cell), complex formation, degradation, and therefore cell signaling networks.

Cell signaling is part of the complex system of communication that controls basic cellular activities and coordinates cell actions. In order to maintain immunity, cell development and tissue repair, it is essential that cells correctly perceive and respond to their microenvironment. Diseases such as cancer, diabetes and autoimmunity are caused by errors in cellular information processing. By understanding cell signaling, diseases may be treated effectively. Systems biology research helps us to understand how changes in these networks may affect the transmission and flow of information.

2.2 Comparing Sequences

Mass spectrometry (MS) is an analytical technique for determining the elemental composition of a sample or molecule. It is also used for clarifying the chemical structures of molecules, such as peptides. The MS principle consists of ionizing chemical compounds to generate charged molecules or molecule fragments and measuring their mass-to-charge-ratios. The MS methodology has been further enhanced by use of tandem MS (MS/MS) where the first MS generates a mass spectrum, a specific ion is selected from the spectrum, fragmented, then used in a second MS to generate a more specific mass spectrum. A second enhancement takes a complementary technology, liquid chromatography (LC), and couples the LC data with MS data to more confidently identify molecules. An LC may also be paired with a tandem MS (LC-MS/MS).

Advances in high-throughput mass spectrometry allow identification of thousands of phosphorylation sites in a single experiment (Beausoleil, et al. [3], Olsen, et al. [4]). A primary objective of the analysis of phosphorylation site data is to discover cellular signaling pathways that give rise to the observed phosphorylated proteins.

Kinase-substrate specificity plays a key role in this discovery process. Consensus sequence motifs recognized by the active site in the catalytic domain of kinases are an important component of substrate specificity in protein phosphorylation Hjerrild, et al. [5]. A first step to assigning specificity is to compare the unknown phosphopeptide data generated by MS to known data in a verified database using a method known as sequence alignment.

A sequence alignment is a way of arranging protein sequences to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Aligned sequences of amino-acids residues are typically represented as rows within a matrix. An example of aligned sequences can be found in chapter 3 of this thesis. Gaps are inserted between the amino-acid residues so that identical or similar sequences are aligned in successive columns.

A sequence motif is an amino-acid sequence pattern that is widespread in a protein sequence and is believed to have biological significance such as controlling biosynthesis, directing a molecule to a specific site within the cell, or regulating its maturation.

Amino acid sequences that are important for protein function and structure change very slowly in a given protein family evolution. These conserved sequence motifs are called consensus sequences. They are a way of representing the results of a multiple

sequence alignment, where sequences are compared to each other for the purpose of discovering the function of a protein by comparing amino acids sequences to that of proteins with known function. For enzymes, the success of this approach is usually due to evolutionary relationships between the structures of active sites that direct a particular biochemistry, such as transfer of a phosphate group to a protein. The structure of active sites frequently produces substrate specificity by requiring a part of the substrate to fit into the active site. Consequently, the sequence context of a phosphorylation site provides clues to which kinase was responsible for the phospho-group transfer.

Recognition properties of the active site alone are not sufficient to uniquely identify physiological substrates of specific kinases. Contextual features also contribute to substrate specificities of protein kinases in vivo. Well documented contextual features include subcellular compartmentalization, colocalization by anchoring proteins and scaffolds, as well as temporal and cell-type specific coexpression [6].

2.3 Data Mining

2.3.1 Data Mining Applications

The knowledge gained from data mining helps in making complex decisions by identifying when such decisions need to be made and validating the rationale behind such decisions. Due to the ever increasing vast amount of data being generated, it is necessary to perform data mining to identify interesting patterns and gain knowledge from the data.

There are numerous approaches to data mining proteomic data. Data mining techniques have been developed to predict proteins from MS/MS spectra, predict proteins from peptide sequences, predict protein function from protein structure, and predict protein function from protein sequences and to search data by generating large databases

and developing tools specific to the data set. Most of the current data mining techniques involve building models to fit the data. These models are usually protein-centric, predicting functions and structures of the entire protein sequence as opposed to an individual peptide.

The question that remains to be answered is: How well do these current data mining techniques and models apply to phospho-proteomics? I will first review proteomics data mining and then focus on phospho-proteomics data mining.

With respect to proteomics, many data mining approaches have been used. The following references describe some approaches applied to proteomic data to gain additional knowledge from the data.

2.3.2 Historical Data Mining for Proteomic Data

Whishart [39] provides a brief overview of techniques that have been used in data mining proteomic data in the last 30 years. Statistical approaches were first used and later the techniques were developed based on information theory. These later techniques include neural networks and Bayesian theory. A more recent technique is to use clustering algorithms. Other popular methods include classification, association and sequence analysis, and regression. Depending on the nature of the data as well as the desired knowledge there is a large number of algorithms for each task.

2.3.3 Data Mining Protein Data Produced from MS

Raj [18] describes an approach to predicting the functions of proteins based on their sequence. It uses existing repositories of protein data Uniprot [63] which contains both protein sequence and functional data and Prosite [64] containing protein sequence and functional data. The method chooses a set of sample data from Uniprot and

combines the data with linking information in the Prosite data. This method results in a set of predictor attributes that are used as input to data mining algorithm, C4.5, described in Quinlan [62]. This algorithm produces comprehensible knowledge that can be easily interpreted in the form of If-Then rules. It allows biologists to be able to validate the knowledge that has been inferred from the data.

Pfaltz, et al. [19] presents a closed set data mining paradigm which is a good approach for uncovering deterministic, causal dependencies when the relationship of the data is dense. Given a closure operator ϕ , a closed system is one that satisfies the three basic closure axioms: $X \subseteq X.\phi$; $X \subseteq Y$ implies $X.\phi \subseteq Y.\phi$; and $X.\phi.\phi = X.\phi$, for all X , Y . This method utilizes an algorithm to incrementally combine closed sets one at a time to mine the associations.

Li, et al. [21] apply data mining techniques to MS data sets to identify serum proteomic patterns that distinguish the serum of ovarian cancer cases from non-cancer controls. It uses a support vector machine-based method and statistical testing and genetic algorithm-based methods are used for feature selection.

Fetrow, et al. [24] use data mining to predict functions of proteins based on their sequence and structure. They use a novel method for identifying protein function by creating descriptors of protein active sites, termed “fuzzy functional forms” or FFFs, for protein active sites that are based on the geometry and conformation of the active site. The FFFs can specifically identify the functional sites of these proteins from their predicted structures.

Huang, et al. [28] use a systems biology approach to study cellular signaling networks and a clustering analysis to better understand the molecular basis of GBM tumor biology and to discover non-intuitive candidates for therapeutic target validation.

Cannataro, et al. [40] developed PROTEUS, a software environment for composing and running bioinformatics applications in heterogeneous, multi-owned environments. It is comprised of two domain ontologies that describe proteomics, PROTON, and data mining, DAMON. Using PROTON, the user can choose appropriate bioinformatics knowledge discovery tools or access protein data banks to conduct protein analysis. Using DAMON, the user can choose appropriate data mining tasks (e.g. classification, clustering, etc.) and software tools, related to the bioinformatics processes described by PROTON. These ontologies allow for the user to simplify the design of bioinformatics applications for specific data sources and purposes.

Cerqueira, et al. [48], Xu, et al. [49], [51], Higdon, et al. [50], and Yates, et al. [56] aim is to improve phosphopeptide/protein identification. Their data mining approach uses a support vector machine classifier for preprocessing MS spectra data. Their approaches take a support vector machine classifier used for determining the useful peaks in a spectrum and train it with the procedure of assigning a peptide sequence.

Halligan [52] describes methods for improving partial phosphopeptide/protein identification of MS data.

Shannon, et al. [54] developed an open source software system for integrating bioinformatic tools and data sources. This system is focused on combining information at the gene level.

Desiere, et al. [55] focuses on annotating the human genome with protein level information.

2.3.4 Data Mining for Phospho-Peptide Data Produced from LC-MS/MS

Puente, et al. [26] describes an approach to data mining phospho-proteomic data. This method combines proteomics and bioinformatics technologies to annotate peptide sequences obtained by LC-MS/MS. They used kinase-substrate interaction databases to reconstruct a kinase signaling network based upon their experimentally identified phosphorylation events.

Bodenmiller, et al. [35] developed a database called PhosphoPep that containing more than 10,000 unique high-confidence phosphorylation sites mapping to nearly 3500 gene models and 4600 distinct phospho-proteins of the *Drosophila melanogaster* Kc167 cell line. It is the most comprehensive phosphorylation map of any single source to date. PhosphoPep also comes with an array of software tools that allow users to browse through phosphorylation sites on single proteins or pathways, to easily integrate the data with other external data types such as protein-protein interactions and to search the database via spectral matching. The data can be exported to use in other methods.

Nakayasu et al. [41] utilize existing databases and tools to annotate phospho-peptide data that was identified using LC-MS/MS. Taken together the author's phospho-proteomic data provide new insights into the molecular mechanisms governed by protein kinases and phosphatases in *T. cruzi*.

Obenauer, et al. [57] describe Scansite, a tool that can be utilized to determine sequence motifs for phosphorylated proteins.

Developing data mining techniques at the peptide level allows for determination of biological significance with respect to each peptide sequence of a protein instead of the entire protein sequence. Each peptide has potential to affect protein function and hence it's signaling pathways.

2.4 Problem Statement /Issues that need to be addressed

A common goal of proteomics is to examine the results of various treatments of biological organisms in order to determine which proteins are statistically significant for a specific treatment such as radiation exposure and to relate these proteins to biological processes, such as cellular signaling pathways.

Changes in protein abundance do not necessarily reflect change in biological activities; however, protein phosphorylation and de-phosphorylation are often associated with cell signaling. In ordinary proteomics, all observed peptides that are associated with the same protein can be used to estimate the abundance of the protein relative to a control. This has the effect of increasing the number of replicates in statistical analysis of significance difference. Since different phosphorylation sites on the same substrate may have a differential biological consequence, spectral count data for phospho-peptides should only be combined if the phospho-peptides contain the same phosphorylation sites.

Another challenge is to extract biological understanding from the statistically significant peptides. In this thesis biological understanding means associating these peptides with cellular signaling processes. Knowing the kinase or phosphatase responsible for phosphorylation or de-phosphorylation of a protein is a key to associating phosphor-peptides with signaling pathways. However, our limited knowledge of kinase

and phosphatase substrate consensus motifs inhibits this approach. This particular issue is partially solved by the NetworKIN tool described in Chapter 3.

There are a number of other issues that need to be addressed when developing tools for data mining phospho-proteomic data. Several issues relate to the size, number, and diversity of data sets available. Data mining involves many attempts to discover associations within a dataset using varying data, parameters, and methods. Manually setting up cases is burdensome so data mining tools need to be automated for repeated use.

The task of generalizing data mining tools for use with diverse data sets is difficult. There are an ever growing number of databases that contain proteomic data. However each of these databases contains its own set of unique identifiers which makes it difficult to combine the knowledge from each database to gain additional knowledge. Hence, each data mining tool is usually developed for a specific type of data in search of a specific type of knowledge.

A final issue that needs to be noted is the limitations of current global sequence alignment and local sequence motif identification. Deriving knowledge from the alignment of sequences is limited by the number of sequences whose function is already known.

This thesis focuses on the development of data mining processes and tools that address the challenges of statistical significance and biological interpretation of phospho-proteome data. These processes and tools were applied to PNNL data of fibroblast skin data set that was exposed to low dose radiation [10] and enabled the association between the data set and the signaling pathways affected by this treatment.

CHAPTER 3 DATA MINING PHOSPHO-PROTEOMIC DATA

3.1 Motivation

Low doses of radiation come from many sources every day. Material from the Earth has uranium decay products, solar flares and cosmic radiation bombard the earth, living at higher altitudes increases exposure, occupational exposure may come from the use of radiation sources for pipe inspection or thickness gauges, and medical exposure from cancer treatment. With the increase in the use of medical imaging, exposure to low doses of radiation from diagnostic tools takes many forms such as dental X-rays, MRI, and CAT scans. Investigating phospho-peptide abundance change due to exposure to low dose radiation will help in the understanding of the health effects of low dose radiation.

The focus of the research in this thesis was on the KDD process, and data mining in specific, which make up enrichment analysis. The purpose of enrichment analysis is to identify the known biological processes for which involvement of proteins with observed differential abundance is unlikely to have happened by chance alone. This is done by identifying the phospho-peptides with statistically significant differential abundance and annotating them with associated cellular signaling networks that are tied to cell functions. PNNL data [10] was generated using human skin fibroblast in tissue culture exposed to low doses of ionizing radiation. The goal of analyzing the differential abundance of phospho-peptides due to low dose radiation is to associate them with cellular signaling and understand the possible impact of low dose radiation on cellular function. An alteration in the abundance of phospho-peptides may result in a very different pattern of cellular signaling thus changing the cellular functions.

3.2 MS Process

The PNNL data [10] was obtained by use of a high sensitivity metal-free automated nanoLC-MS platform specifically designed for phospho-peptide analyses. The platform includes high sensitivity LC-MS/MS, intact phospho-protein analysis and bioinformatics tools to facilitate accurate identification and quantification of phospho-peptides. Spectral count and ion-current peak area are the spectral features most commonly used to assay relative peptide abundance. Spectral count, the number of MS/MS spectra containing a feature associated with a given peptide, is the experimental measurement used by Stenoien and coworkers [10] to assay altered phospho-peptide expression due to 2 and 50 cGy of X-ray exposure. For each biological sample, (sham irradiated, 2 cGy and 50 cGy exposures) 4 injections of enhanced phospho-protein were subjected to LC-MS/MS analysis.

3.3 The Data Set

In the original dataset, more than 7100 phospho-peptides were detected in at least one of the 12 separate MS runs. The data was supplied in an Excel spreadsheet containing the spectral counts observed for a given peptide in each MS run together with its amino acid sequence and information about the protein database entry associated with the detected phospho-peptide. The research described in this thesis focused on determining which of the detected peptides were differentially expressed due to radiation exposure and performing analysis aimed at revealing the biological processes responsible for differential expression.

Since a spectral count equal to zero was observed in many cases, the first step in processing the MS data was to reduce the size of the data set by removing phospho-

peptides observed in only one MS/MS spectrum. The justification for removing the single occurrence phospho-peptides is that they are likely to be false positives. Also, the single occurrence data is insufficient to test for differential expression due to radiation exposure. The reduced data set contained 3020 phospho-peptides.

3.4 Overview of Knowledge Discovery Process Branches

The first step in my analysis was the identification of the statistically significant phosphor-peptides in exposed samples relative to sham irradiated controls. The process of generating phospho-peptides for the MS analysis breaks the phosphorylated protein at varying points in its amino acid sequence by trypsin. Sometimes the resulting phospho-peptides are biologically equivalent because they contain the same phosphorylation site(s) of the substrate. To enhance the power of tests for statistical significance, I combined spectral count for equivalent phospho-peptides.

Once the statistically significant peptides were identified, two approaches were used to link them with signaling pathways. The first method started with knowledge of cellular signaling processes to find phospho-peptides associated with specific signaling pathways and then to ask if any of the identified peptides matched with PNNL data [10]. I refer to the first method as the “signaling pathways to data” approach. The diagram of this method is shown in Figure 3.1. The method of starting with the known models and data as the goal and building a chain of reasoning backwards to experimental data shows a chain of reasoning similar to that established by a backward reasoning expert system.

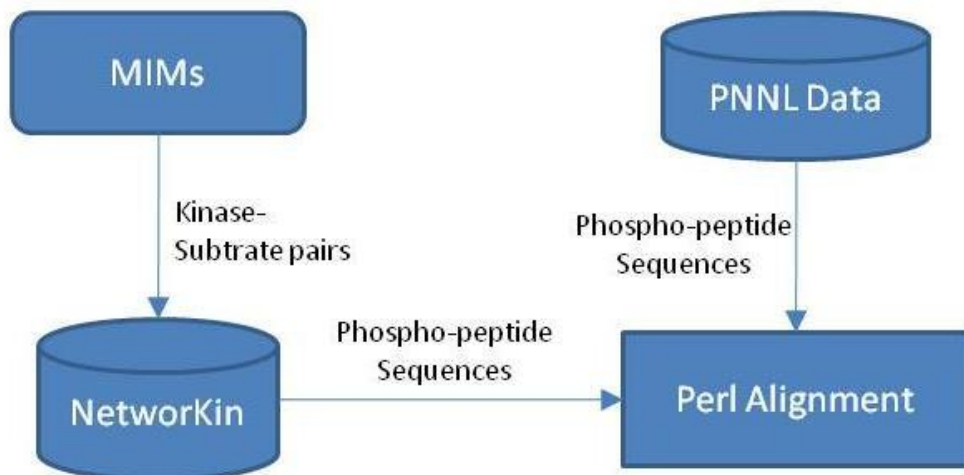


Figure 3.1 Signaling Pathways to Data Method

The second method started with differentially expressed peptides in PNNL data [10] and worked to identify association of these peptides with signaling pathways (“data to signaling pathways”). A diagram of this method is shown in Figure 3.2.

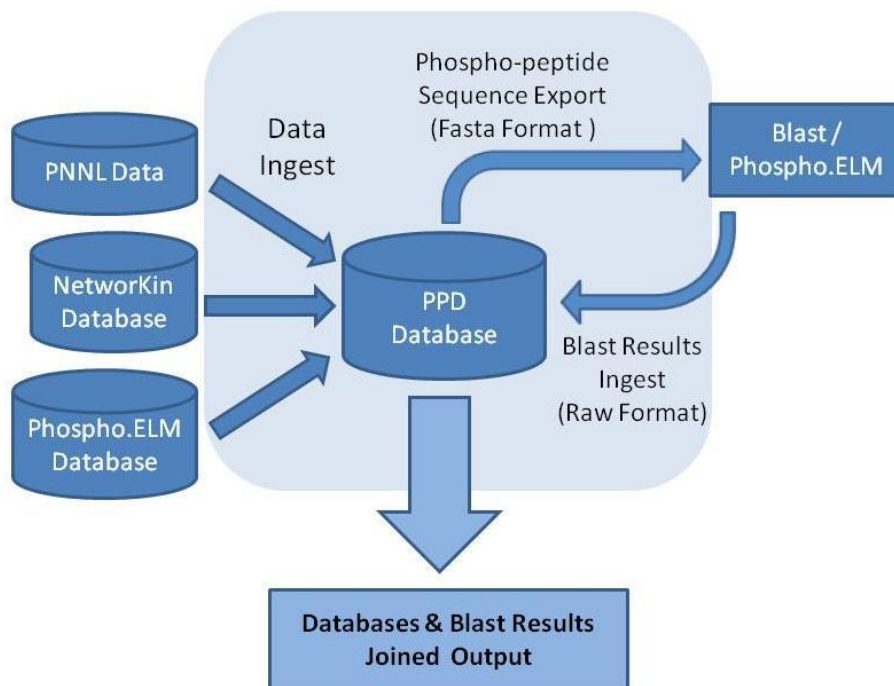


Figure 3.2 Data to Signaling Pathways Method

By working in stages, my reasoning was breadth first – find all the relationships at one level before going to the next reasoning step. The subsequent stages involve finding the relationships that exist from the experimental data to existing models and data.

3.5 Identifying Statistically Significant Data – Combining Peptides

To identify statistically significant phospho-peptides, I built a Microsoft Access relational database, hereafter referred to as PPD database, to automatically combine spectral counts of equivalent phospho-peptides. A relational database is a collection of data items organized as a set of tables from which data can be accessed or reassembled in many different ways without having to reorganize the database [65]. The resulting groups of data are organized into related groups and are therefore much easier for people to understand. Access combines the database with a graphical user interface and software development tools.

3.5.1 Combining Spectral Count Data

To facilitate the discussion of my PPD database approach for finding equivalent phospho-peptides, I introduced a “*Trivial*” case, where peptides with the same substrate that differ by at most one amino acid at the beginning, end, or both, are considered a group of phospho-peptides to be combined. Prior to examining the data, it seemed possible for a phospho-peptide to have 3 longer *Trivial* siblings, and each of those to have 3 longer *Trivial* siblings, and so on. If such a pattern existed, I planned to explore whether the collection of *Trival* groups based on the “shortest” phospho-peptide in the collection should be combined. The basis for exploring the *Trivial* case was the variability in the results from the trypsin digestion of phospho-peptides. The resulting

detected phospho-peptides from the trypsin digestion may or may not contain the cut sites at lysine and arginine.

In the query (Figure 3.3) to find the trivial cases in PNNL data [10], the data is pulled from two PPD database tables. The first table, PPD database PrimaryTable, is the original data set with a row Id added as a unique surrogate key that also allowed tracing back to the original data row, should such a step be required. The relational database model was initially developed with the assumption that each row is unique. Adding the row Id insured row uniqueness regardless of the row uniqueness of the original data.

The original data set contains phospho-peptide sequences, each with its associated substrate, protein, spectral counts, field descriptions, and links to other protein databases. A copy of PPD database PrimaryTable containing only three columns, row Id, peptide sequence, and substrate, is referred in the query as SecondaryTable and will be referred to in the following discussion as PPD database SecondaryTable. The narrower table is a projection of the original table and is more efficient to query. The average record size in the PPD database PrimaryTable is 135 bytes. The average record size in PPD database SecondaryTable is 53 bytes. The narrower table is 58% faster to read from disk than the original table. When the extended information from PPD database PrimaryTable was needed as part of an output to a later stage of processing, PPD database PrimaryTable could be included in the final query and the extended information added to the output.

```

SELECT SecondaryTable.id AS SecondaryTable_id, SecondaryTable.peptide
AS SecondaryTable_peptide,
    PrimaryTable.ID AS PrimaryTable_id, PrimaryTable.peptide AS
    PrimaryTable_peptide, SecondaryTable.substrate AS
    SecondaryTable_substrate, PrimaryTable.substrate AS
    PrimaryTable_substrate
FROM SecondaryTable INNER JOIN PrimaryTable ON
SecondaryTable.substrate = PrimaryTable.substrate
WHERE (((InStr([PrimaryTable].[peptide],[SecondaryTable].[peptide]))>0) AND
    ((Len([PrimaryTable].[peptide]))>Len([SecondaryTable].[peptide])))
ORDER BY SecondaryTable.id, PrimaryTable.ID;

```

Figure 3.3 PPD database query initial *Trivial* group definition

The *Trivial* method was initially based on the definition of a proper substring where string *a* is a proper substring of string *b* to explore the possibility of collections of *Trivial* cases based on a “shortest” phospho-peptide. In this situation I am looking for a pair of peptides such that their string descriptions hold a proper substring relationship: the pair of peptides (*a*, *b*) such that *a* is an element of list of peptides with substrate *c* in PPD database SecondaryTable, *b* is an element of the list of peptides in PPD database PrimaryTable with substrate *d* where *c* = *d* and *a* is a substring of *b* and the length of *a* < length of *b* (proper substring). Assume the following two example tables, Table 3.1 and Table 3.2 represent the peptide/substrate content of both PPD database PrimaryTable and PPD database SecondaryTable in the PPD database that I wish to examine for *Trivial* groups.

Id	Peptide	Substrate
93	AEQGS*EEEGEGEEEEEEGGESK	ABCF1
94	AEQGS*EEEGEGEEEEEEGGESKADDPYAHLSK	ABCF1
95	AEQGS*EEEGEGEEEEEEGGESKADDPYAHLSKK	ABCF1
2278	KAEQGS*EEEGEGEEEEEEGGESK	ABCF1

Table 3.1 Example data for initial *Trivial* group definition

The initial definition resulted in phospho-peptide Id 93 being a proper substring of phospho-peptides Id 94, 95, and 2278 while phospho-peptide Id 2278 is not a proper substring of phospho-peptides Id 94 and 95. The significant difference in length between phospho-peptides Id 94 and 95 and phospho-peptides Id 93 and 2278 was not the expected one or two amino acid difference. As a result, the definition of a *Trivial* group was revised to be a pair of phospho-peptide sequences that differ by the addition of just one amino acid to the beginning or end of the “shortest” phospho-peptide. Using this definition, phospho-peptides Id 93 and 2278 are a *Trivial* group and phospho-peptides Id 94 and 95 are a *Trivial* group.

Id	peptide	Substrate
2593	KS*LDSDES*EDEEDDYQQK	PDAP1
2594	KS*LDSDES*EDEEDDYQQKR	PDAP1
4737	S*LDSDES*EDEEDDYQQK	PDAP1
4738	S*LDSDES*EDEEDDYQQKR	PDAP1

Table 3.2 Example data for revised *Trivial* group definition

The revised definition led to chains of *Trivial* groups for the same substrate. Assume that the set $[a, b, c, d, e, f]$ are peptides with the same substrate, and the following set of *Trivial* groups exists: $[(a,b), (a,c), (a,d), (b,d), (b,f), (c,d), (d,e)]$. The purpose of this exercise was to explore what it would mean if various patterns or chains of *Trivial* groups existed. When viewed as edges of a graph where peptides are nodes and *Trivial* groups are edges, the nodes and edges form a directed acyclic graph (Figure 3.4). From Table 3.2, peptide Ids 4737, 4738, 2593, and 2594 may be mapped to $a, b, c,$ and d in Figure 3.4. The data was observed to hold various combinations of *Trivial* groups. However it was observed in the data that nodes d and f were mutually exclusive and node

e never occurred in the PNNL data [10]. The final definition of *Trivial* group allowed only edges (a,b) and (a,c) . If other edges existed, they would be part of a different *Trivial* group.

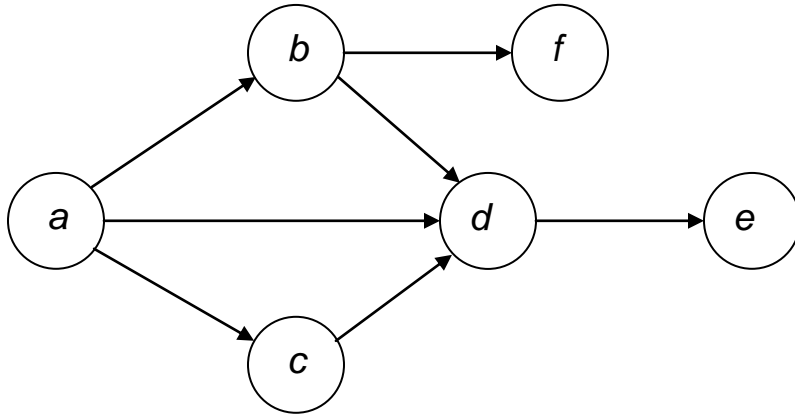


Figure 3.4 Directed acyclic graph of potential *Trivial* groups

The formal definition of *Trivial* groups is the set of peptides pair (a,b) such that a is a peptide in PPD database SecondaryTable with substrate, b is a peptide in PPD database PrimaryTable with substrate d where $c = d$ and a is a substring of b and the $(\text{length of } a) = (\text{length of } b)+1$.

The general *Phosphorylation site match* case extends the *Trivial* case to include nodes a , b , c , and d in Figure 3.4. A complex SQL query was developed to insure that both the count of phosphorylation sites matched and one peptide name was a substring of the other peptide name. A peptide name matching itself was allowed. As with the previous methods, substrate also had to match. The SQL query used was:

```

SELECT SecondaryTable.id AS SecondaryTable_id, SecondaryTable.peptide AS
SecondaryTable_peptide,
SecondaryTable.substrate AS SecondaryTable_substrat, PrimaryTable.ID AS
PrimaryTable_id,
PrimaryTable.peptide AS PrimaryTable_peptide, PrimaryTable.substrate AS
PrimaryTable_substrat, If(InStr(SecondaryTable.peptide,'*')>0,
If(InStr(InStr(SecondaryTable.peptide,'*')+1,SecondaryTable.peptide,'*')>0,
If(InStr(InStr(InStr(SecondaryTable.peptide,'*')+1,SecondaryTable.peptide,'*')+1,Second
aryTable.peptide,'*')>0,
If(InStr(InStr(InStr(InStr(SecondaryTable.peptide,'*')+1,SecondaryTable.peptide,'*')+1,Se
condaryTable.peptide,'*')+1,
SecondaryTable.peptide,"*")>0,4,3),2),1),0) AS star_count INTO
2ndResultwithsubstrateandpeptidematch
FROM SecondaryTable, PrimaryTable
WHERE (((SecondaryTable.substrate)=PrimaryTable.substrate) And
((InStr(PrimaryTable.peptide,SecondaryTable.peptide))>0) And
((Len(PrimaryTable.peptide))>Len(SecondaryTable.peptide))) And
If(InStr(SecondaryTable.peptide,'*')>0,
If(InStr(InStr(SecondaryTable.peptide,'*')+1,SecondaryTable.peptide,'*')>0,
If(InStr(InStr(InStr(SecondaryTable.peptide,'*')+1,SecondaryTable.peptide,'*')+1,Second
aryTable.peptide,'*')>0,
If(InStr(InStr(InStr(InStr(SecondaryTable.peptide,'*')+1,SecondaryTable.peptide,'*')+1,Se
condaryTable.peptide,'*')+1,
SecondaryTable.peptide,"*")>0,4,3),2),1),0)=
If(InStr(PrimaryTable.peptide,'*')>0,
If(InStr(InStr(PrimaryTable.peptide,'*')+1,PrimaryTable.peptide,'*')>0,
If(InStr(InStr(InStr(PrimaryTable.peptide,'*')+1,PrimaryTable.peptide,'*')+1,PrimaryTable.
peptide,'*')>0,
If(InStr(InStr(InStr(InStr(PrimaryTable.peptide,'*')+1,PrimaryTable.peptide,'*')+1,Primary
Table.peptide,'*')+1,
PrimaryTable.peptide,"*")>0,4,3),2),1),0)
ORDER BY SecondaryTable.substrate, SecondaryTable.id;

```

The *If* calls in the query have the function of counting the number of “*”s found in the *peptide* field. If one “*” is found (the location in the string is a positive integer), the true clause looks for another “*” starting from the position after the found “*”. The true clauses are nested to look for up to four “*” in the *peptide* name string. This level of nesting was found to be sufficient for the data set. The *Where* clause of the query has two *If* structures, one for the *peptide* name from PPD database SecondaryTable, the other for the *peptide* name in PPD database PrimaryTable, to assure the count of the phosphorylation sites in each sequence match.

The *If* nesting approach was required by the SQL language specification. The SQL language is based on the concept of sets. The manipulation of sets may be described using simple algebraic operators. Recursive operators or functions calls are not part of the SQL language. Many current databases that implement the SQL language also provide a vendor specific procedural programming language that may be used to define new functions. The procedural programming works counter to the set-based query processing and optimizations of SQL. User defined functions are applied iteratively on each row in the set of rows. While a database is designed to know how to optimize build-in functions, user-defined functions must be treated as unknown code making them difficult to optimize and leading to poor performance when they are used.

A review of the resulting table showed some phospho-peptide names matched with up to four shorter names. An example is included in Table 3.3:

Secondary Table_id	SecondaryTable_peptide	Primary Table_id	PrimaryTable_peptide	Substrate
1372	GAGDGS*DEEVDGKADGAEAKPAE	2352	KGAGDGS*DEEVDGKADGAEAKPAE	MYH9
2350	KGAGDGS*DEEVDGK	2352	KGAGDGS*DEEVDGKADGAEAKPAE	MYH9
2351	KGAGDGS*DEEVDGKADGAEAK	2352	KGAGDGS*DEEVDGKADGAEAKPAE	MYH9
1371	GAGDGS*DEEVDGK	2352	KGAGDGS*DEEVDGKADGAEAKPAE	MYH9

Table 3.3 Example group with site match and substring

In this example, all the longer peptide name strings have a proper substring in peptide id 1371. However, five cases were found where two SecondaryTable_peptide names from PPD database SecondaryTable associated with a common PrimaryTable_peptide name from PPD database PrimaryTable did not have a proper substring relationship. Instead, they had an overlapping string relationship.

ID	Peptide	Common ID	Common_Peptide	Substrate
1504	GGVT*GSPEASISGSKGDLK	1551	GKGGVT*GSPEASISGSKGDLK	AHNAK
1550	GKGGVT*GSPEASISGSK	1551	GKGGVT*GSPEASISGSKGDLK	AHNAK
493	ASS*HSS*QTQGGGSVTKK	1700	GRASS*HSS*QTQGGGSVTKK	LMNA
1699	GRASS*HSS*QTQGGGSVTK	1700	GRASS*HSS*QTQGGGSVTKK	LMNA
960	EGEPTVY*SDEEHPKDESAR	2948	LLKEGEEPTVY*SDEEHPKDESAR	PGRMC1
2947	LLKEGEEPTVY*SDEEHPK	2948	LLKEGEEPTVY*SDEEHPKDESAR	PGRMC1
63	ADS*GEEENTKNGGEK	4086	RADS*GEEENTKNGGEK	CXorf26
4085	RADS*GEEENTK	4086	RADS*GEEENTKNGGEK	CXorf26
1520	GHYEVT*GSDDETGKLQGSVSLASK	5264	SKGHYEVT*GSDDETGKLQGSVSLASK	AHNAK
5263	SKGHYEVT*GSDDETGK	5264	SKGHYEVT*GSDDETGKLQGSVSLASK	AHNAK

Table 3.4 Example group with site match, substring, and overlap

For example peptide Id 1550 and 1504 are proper substrings of peptide 1551. The relationship shown in Table 3.4 between peptide Id 1550 and 1504 shows that they overlap on the string GGVT*GSPEASISGSK with a prefix of GK and postfix of GDLK.

The proper substring relationships and the overlap relationships were handled in two steps. In the following steps, it is assumed that all peptides involved have a common substrate.

The first step involves five passes of SQL statements to collect the transitive substring relationships that exist. Recall that the SQL language does not support recursive functions. Five steps of defining transitive pairs were sufficient for the data set. For a set of substring relationships [(a,a), (a,b), (b,b), (b,c), (c,c), (c,d), (d,d), (d,e), (e,e)], the process generates the transitive closure of the set: [(a,a), (a,b), (a,c), (a,d), (a,e), (b,b), (b,c), (b,d), (b,e), (c,c), (c,d), (c,e), (d,d), (d,e), (e,e)]. The peptide contained in all other peptides (“shortest”) is considered the top of the sequence and its peptide id is arbitrarily assigned as the group id for the *Phosphorylation site match (PSM)* group.

The second step then reviews the results to determine if any peptide has been assigned to more than one *PSM* group. Based on the content of the data set, overlapping

relationships are then handled by identifying the common peptide id and arbitrarily assigning its peptide id as the new group id to all peptides associated with the common peptide. An overlap relationship was observed to be not very common. The key constraint in an overlap relationship is that all phosphorylation sites must occur in the overlap section. To see this, assume that either one or the other of the prefix or postfix sections contain a phosphorylation site, but not both. Then if the overlap section has n phosphorylation sites, then one peptide has n phosphorylation sites while the other has $n+1$ phosphorylation sites. The common peptide then has $n+1$ phosphorylation sites. The peptide with n phosphorylation sites violates the constraint that it has the matches the number of phosphorylation sites in the longer peptide. If both prefix and postfix have phosphorylation sites, then the common peptide has $n+2$ phosphorylation sites and both overlapping peptides violate the constraint that they match the number of phosphorylation sites in the longer peptide. Therefore all phosphorylation sites must occur in the overlapping section.

The results of the *PSM* grouping were combined with the extended information in PPD database PrimaryTable and exported to Excel for a presentation to domain experts. Excel is a tool used by the domain experts to review and manipulate data. The domain experts could review groups of phospho-peptides, examine associated attributes, and quickly determine if the groups seemed appropriate and correct for the group definition being used. A sample of rows is included in Appendix B. To facilitate review, the group id was used as part of a row coloring macro to highlight the groups. Groups with a single element were left in white (e.g. Appendix B, rows 2-7), while groups with multiple elements were colored from a rotating palette. The peptide names were aligned on the

first phosphorylation sites and centered in the cell to facilitate visual inspection of the group members, which were associated with the phosphorylation site in the substrate for further analysis. The use of color to identify groups was well received by the domain experts.

Another issue in phospho-proteomics is the fact that the MS can not distinguish all phospho-peptides because some may have the same mass-to-charge ratio. An example of this is the phospho-peptide sequences STPFIVPS*SPTEQEGR and STPFIVPSS*PTEQEGR, where the phosphorylation site is noted by an *.

The process of identifying indistinguishable phospho-peptides in PNNL data [10] begins replacing all the phosphorylation sites, “*”, with the empty string “”. In SQL, the definition of *peptide_nosite* is:

```
SELECT Replace([peptide], '*', '') AS peptide_nosite
FROM SecondaryTable
```

Let $f_{peptide_nosite}$ be defined as the function that takes a *peptide* name and returns a *peptide_nosite* name. The second part of the group definition addresses membership and includes identical strings but excludes all proper substrings and overlapping strings. A *Removal of PSM* group is defined as the set of phospho-peptides from PPD database PrimaryTable such that for all phospho-peptides a, b in the PPD database PrimaryTable and c is the substrate for phospho-peptide a and d is the substrate for phospho-peptide b , $a \not<b, c = d, f_{peptide_nosite}(a) = f_{peptide_nosite}(b)$. The unique *group_id* for the *Removal of PSM* group will be the row id from the alphabetically ordered table of unique *peptide_nosite* names. Table 3.5 shows how phospho-peptides

with a variety of phosphorylation site locations would reduce the same phospho-peptide if phosphorylation sites are ignored.

Group_id	peptide_nosite	ID	peptide	substrate
70	ASSHSSQTQGGGSVTK	425	AS*SHSSQTQGGGSVTK	LMNA
70	ASSHSSQTQGGGSVTK	494	ASS*HSSQTQGGGS*VTK	LMNA
70	ASSHSSQTQGGGSVTK	495	ASS*HSSQTQGGGSVTK	LMNA
70	ASSHSSQTQGGGSVTK	507	ASSHS*S*QTQGGGSVTK	LMNA
70	ASSHSSQTQGGGSVTK	508	ASSHS*SQTQGGGSVTK	LMNA
70	ASSHSSQTQGGGSVTK	510	ASSHSS*QTQGGGSVTK	LMNA
70	ASSHSSQTQGGGSVTK	511	ASSHSSQT*QGGGSVTK	LMNA
70	ASSHSSQTQGGGSVTK	512	ASSHSSQTQGGGS*VTK	LMNA
70	ASSHSSQTQGGGSVTK	513	ASSHSSQTQGGGSVT*K	LMNA

Table 3.5 Example group with site removed and exact match

The spectral-count observations were totaled for all members in a group of equivalent phospho-peptides and the total was assigned to one of the group members as described above: the “shortest” or the common overlap peptide. Similarly, spectral counts of indistinguishable peptides were combined.

3.5.2 Statistical Significance

After combining the spectral counts, the next step was to determine which phospho-peptides in irradiated samples were significantly different compared to the sham control. The key term in the previous statement is “significant.” There are several methods of statistical significance that might apply to the data set to determine which peptide sequences were differentially expressed

Statistical methods used in comparative quantification of proteins detected in label-free MS experiments were recently reviewed by Wong et al. [12]. Even combining spectral counts for equivalent peptides, a large number of cases with a spectral count of zero remained making it difficult to use statistical tests that utilize multiple technical reps.

Instead, I treated the sum of spectral counts in the four MS technical reps on a given biological samples as non-replicated spectral count data.

Statistical tests for non-replicated data are generally of the “goodness-of-fit” type, which test the null hypothesis that observations conform to a model prediction. The model of interest is that the abundances of phospho-peptides in treated and control samples are not different; hence, their spectra count should be the same. Zang et al. [13] compared several tests for statistical significance of non-replicated spectral-count data and concluded that the G-test was best among them due its computational simplicity and ease of extension to more than 2 classes. Old et al. [14] compared the use of peptide ion intensity to spectral count for quantitative comparison of MS data. They reported that both experimental techniques were capable of distinguishing protein abundance changes of at least 2.5 fold.

A spectral count of zero causes numerical problems in execution of the G-test, which involves terms of the form $O \ln(O/E)$ where O is the observed spectral count and E is the model prediction. It is general practice to avoid these problems by adding a constant to all of the spectral count data (f-factor or correction factor). There are several sources addressing which statistical tests are best for a data set Sokal, et al. [11], Zhang, et al.[13], McDonald[66]. However there is little discussion about the use of correction factors for data that could cause problems in the execution of the G-Test. Old[14] states that he predetermined that an optimal correction factor for his data set was 1.25. Beissbath et al. [15] used several values including .5 and 1.25. It appears that it is really up to the owner of the data set to determine the value for the correction factor f that results in the most accurate G-Test values. Selecting a smaller value, results in a greater

number of phospho-peptides that are considered to be statistically significant.

Determining whether or not these resulting statistically significant phospho-peptides are truly significant with regards to affecting protein function, is left up to experimental verification. For my thesis, several values were tried for the correction factor f .

The G-test can be implemented using additional PPD database queries.

Application of the G-test, using an f -factor of 1.25, to the 2 cGy data revealed 87 phospho-peptides with abundance significantly different from control at 95% confidence. Similarly, 37 phospho-peptides in the 50 cGy data were predicted to have abundance significantly different from control. See Appendix D for a complete list of statistically significant phospho-peptides. Using an f -factor of .5 resulted in approximately twice as many statistically significant phospho-peptides for both the 2cGy and 50cGy treatments. Phospho-peptides observed in treated samples to have abundance significantly different from control were retained for analysis of possible kinases that could be responsible for the observed substrates.

3.6 Signaling Pathways to Data

The next question to ask is what is the biological meaning of the phospho-peptides found to be statistically significant with respect to the low dose treatment?

One approach to address this question was referred to in section 3.4 as *Signaling Pathways to Data*. Molecular Interaction Maps (MIMs) [58] contain known interactions between proteins including kinase-substrate interactions that induce phosphorylated sites. I manually searched MIMs to obtain a list of kinase-substrate pairs at phosphorylation sites. From the manual search, approximately 94 kinase-substrate pairs were identified from the *Chk2*, *EGFR*, and *p53 and Mdm2* MIMs found at [61]. The MIM for *Chk2* is

shown in Figure 3.5. The maps are generated as vector diagrams and are difficult to include as figures in publications. The highlighted rectangles in the map represent kinases and substrates. The blue arrows with a “p” mark a phosphorylation site.

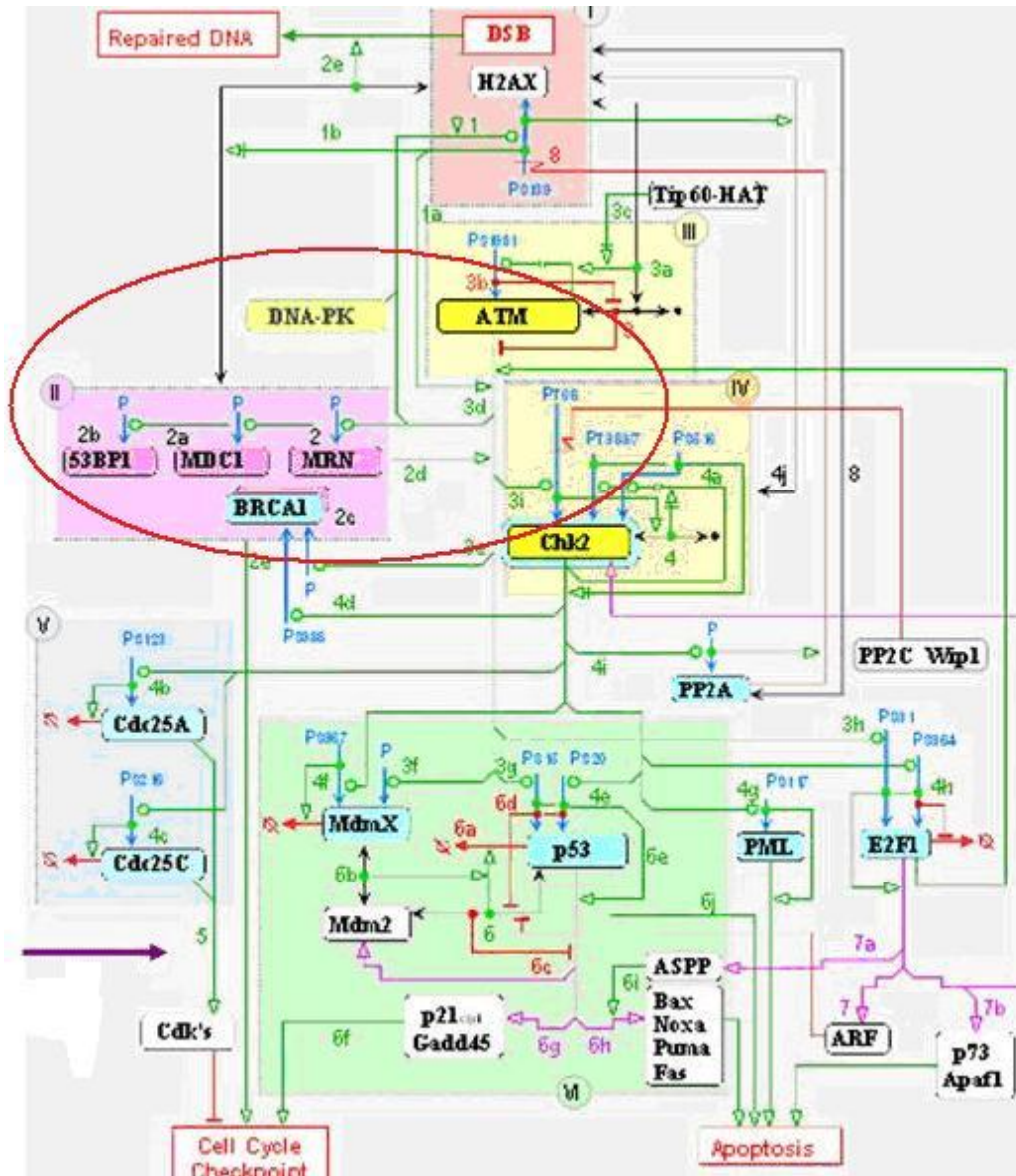


Figure 3.5 Molecular Interaction Map (MIM) for Chk2

The next step was to take the kinase-substrate data from the three MIMs and use each kinase-substrate pair as a search term in the NetworKIN algorithm[8], which predicts a list of phospho-peptides likely to be associated with a given kinase-substrate pair. This algorithm augments consensus motifs with biological context for kinase and phosphatase interactions with proteins. Site specific kinase-substrate predictions of NetworKIN were obtained from the web interface located at [60].

The NetworKIN website allows users to browse/search and investigate predictions made using the NetworKIN algorithm. The site draws on the latest version of the phospho-proteome in Phospho.ELM [7]. The user may also submit their own kinase and obtain new NetworKIN predictions. The features of NetworKIN are discussed further in Linding et al. [8]. The web interface for NetworKIN is illustrated in Figure 3.6.

NetworKIN found two phospho-peptides in its database generated by the action of kinase AKT1 on substrate ACN1. The results may be exported to a variety of file formats. Tab separated values (TSV) were used in this work. As can be seen in Figure 3.6, NetworKIN contains associated phospho-peptide sequences as well as site context scores, string network and other information. The phospho-peptide sequence is printed in upper case letters with phosphorylation sites marked in lower case letters. In the case of Figure 3.6 the phosphorylation site is an “s” in both results lines.

NetworKIN Home Download License Help Contact

Query Existing Data Submit Phosphoprotein Network Map Statistics

Substrate gene: ACIN1 AND kinase gene: AKT1 Search or try Advanced Search

Export results to: .TSV Export

Select rows by clicking the checkboxes in the first column, then press Show STRING Context to construct a STRING network for the selected entries.

All	Substrate	Position	Context Score	Kinase	PhosphoPeptide	Kinase Family	Motif Score	Short Description Substrate	Source	Shortest Path
<input type="checkbox"/>	ACIN1	S1161	0.790	AKT1	ERRERTR _s EREWD _{RD}	PKB	0.806	Apoptotic chromatin condensati ...	pSITE	String Graph
<input type="checkbox"/>	ACIN1	S1331	0.790	AKT1	HRSRSR _s TPVD _{RG}	PKB	0.676	Apoptotic chromatin condensati ...	pELM pSITE	String Graph

SAMUEL LUNENFELD RESEARCH INSTITUTE MOUNT SINAI HOSPITAL KOCH INSTITUTE for Integrative Cancer Research at MIT EMBL

(C) 2005-2008 Rune Linding and Lars Juhl Jensen
Web-development: Adrian Pasulescu and Marina Ohovsky

GET FIREFOX W3C CSS 2 W3C XHTML 1.0 POSTGRES POWERED BY APACHE

Figure 3.6 Example of NetworKIN output for kinase AKT1 and substrate ACN1

Only a small portion of the kinase-substrate pairs, 28 out of 94 pairs obtained from the three MIMs, were found in the NetworKIN database. These 28 pairs resulted in only 43 predicted phospho-peptides from NetworKIN. Knowing that the substrate found in the MIMs was phosphorylated, I also queried NetworKIN for other kinases that may not be on the MIMs but could also be responsible for the substrate phosphorylation. A total of 302 phospho-peptides were obtained from NetworKIN using the 28 kinase-substrate pairs. Upon comparison, twenty one of the 124 statistically significant phospho-peptides in PNNL data [10] were found to be exact matches with phospho-peptides obtained from NetworKIN. The low level of correlation reflects the limited amount of information available in a specific area when compared to the breadth of the problem space. Nevertheless, to implement *the signaling pathways to data* approach, I determined if any of the phosphorylation sites associated with the 28 kinase-substrate pairs were among the phospho-peptides in PNNL data [10]. To implement this search, I

downloaded and imported the NetworKIN data into a local PPD database and developed a string alignment program in Perl to compare phosphor-peptides in the NetworKIN database with those in PNNL data [10]. The alignment implementation program is included in Appendix A. The alignment program implements the Needleman-Wunsch algorithm [35] for global alignment of sequences. Developed in 1970 by Saul Needleman and Christian Wunsch, this dynamic programming algorithm is commonly used in bioinformatics to align protein sequences. The first step in the global alignment dynamic programming approach is to create a matrix with $M + 1$ columns and $N + 1$ rows where M and N correspond to the size of the sequences to be aligned. Since this example[67] assumes there is no gap opening or gap extension penalty, the first row and first column of the matrix can be initially filled with 0. However, my implementation modified the algorithm to use an Affine gap penalty [38] creating a preference for inserting longer gaps over many separate small gaps.

	G	A	A	T	T	C	A	G	T	T	A
G	0	0	0	0	0	0	0	0	0	0	0
A	0	1									
T	0										
C	0										
G	0										
A	0										

The alignment matrix is filled in based on values in adjacent locations in the matrix. After the matrix fill step, the maximum alignment score for the two test sequences is 6.

		G	A	A	T	T	C	A	G	T	T	A
0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	6

The traceback step determines the actual alignment(s) that result in the maximum score. Note that with a simple scoring algorithm such as one that is used here, there are likely to be multiple maximal alignments. Traceback takes the current cell and looks to the neighbor cells that could be direct predecessors. This means it looks to the neighbor to the left (gap in sequence #2), the diagonal neighbor (match/mismatch), and the neighbor above it (gap in sequence #1). The algorithm for traceback chooses as the next cell in the sequence one of the possible predecessors. In this case, the neighbors are marked in red. They are all also equal to 5.

Continuing on with the traceback step, we eventually get to a position in column 0 row 0 which tells us that traceback is completed. One possible maximum alignment is :

		G	A	A	T	T	C	A	G	T	T	A
0	0											
G		1										
G			1									
A				2	2							
T					3							
C						4	4					
G								5	5	5		
A												6

Giving an alignment of:

```

G A A T T C A G T T A
|   |   | |   |   |
G G A _ T C _ G _ _ A

```

The only alteration allowed to a sequence was the adding of one or more gaps to either query or target sequences. The goal of the search is to find the best possible match between the sequences. The Needleman-Wunsch algorithm was appropriate since I wanted to compare and align the entire (global) sequence. The alignment program utilizes a modified Blosum 62 scoring matrix [37] to determine which match is best. The scoring matrix gives a value to assign to a comparison of two peptides. Since the phospho-peptide sequences in PNNL data [10] had phosphorylation sites in the peptide sequences are marked with an asterisk, I added an additional row and column to the scoring matrix to assign a score for the asterisk. Since the NetworKIN data contained lower case letters to mark phosphorylation sites (see Figure 3.6). I converted the lower case letters marking phosphorylation sites in the NetworKIN data to an asterisk to be consistent with the PNNL data [10].

My alignment code reads three inputs: the scoring matrix, a query set of sequences to search for, and a target set of sequences to be searched. The sets of sequences were contained in text files, one sequence per line. The set of phospho-peptides sequences from the PNNL data [10] was used as my query set of sequences to search for, while the sequences obtained from NetworKIN were the target set of sequences to search for matches.

Initially, all 7100 phospho-peptides in PNNL data [10] were used to search the NetworKIN target set of sequences. The search took about 8 hours. Subsequent searches

used only the statistically significant phospho-peptides to reduce the run time. The alignment code was refined to first search for an exact match where the target sequence is a substring in the query sequence. An exact match, as shown in Table 3.6, means every character in the query phospho-peptide sequence matches some sequence from the target data set including the phosphorylation site(s).

Query String	LAEDEGDS*EPEAVGQSR
Target String	LAEDEGDS*EPEAVGQ

Table 3.6 Successful Exact Match Comparison Result

If no exact match was found, then the alignment code provided a list of the top ten scoring sequences from the searched data. In Table 3.7, the top ten scoring sequences for TRHS*PT*PQQSNR are shown.

Top Sequence	Aligned Sequences	Score
Query 1	--TRHS*PT*PQQSNR-	60
Target 1	PKTRHS-PT*PQQSNRT	
Query 2	----TRHS*PT*PQQSNR	53
Target 2	PPPKTRHS*PT-PQQSN-	
Query 3	--T-R-HS*PT*PQQSNR	28
Target 3	SPTPRSHS*PS-ASQSG-	
Query 4	--TR--HS*PT*PQQSNR-	28
Target 4	PKRRVSHS*PP-PKQ--RS	
Query 5	--TRHS*PT*PQ-QSNR	27
Target 5	PNKRHS-PS*PRPRAPQ	
Query 6	TR-HS*PT-*PQQS-NR	26
Target 6	TRLPS-PTS*PFSSLSQ	
Query 7	-TRHS*P-T*-PQQSNR	26
Target 7	RSR-S-PSHT*RPRRRHR	
Query 8	-TR-HS*PT*PQQSN-R	25
Target 8	DERMHL-PS*PTDSNFY	
Query 9	----TRHS*PT*PQQSNR-	25
Target 9	PQPKNRHS*PS-P-RP-RA	
Query 10	----TRHS*PT*PQQSNR	24
Target 10	TGVATTQS*PT-P-RSHS	

Table 3.7 Top ten sequence alignments for query string TRHS*PT*PQQSNR

The alignment algorithm allowed gaps (“-“) to be inserted to match any character in the other sequence. Both query and target sequences could have gaps inserted to find an alignment. The scoring matrix was refined by numerous trials to generate rankings, which seemed reasonable on inspection. The score is not based on an absolute scale but is only designed to reward matches and penalize mismatches and gaps. The greater the number of matching characters, the higher the score. In Table 3.7, the top scoring alignment is an exact match except for the fact that the query sequence is doubly phosphorylated. The tenth sequence returned has a score of 24 which is a poor match by visual inspection.

PNNL data [10] on TP53BP1 provides an example of the *signaling pathways to data* approach. Exposure to low dose radiation is likely to induce DNA damage. ATM is a well known as a kinase involved in the DNA damage and repair networks and the Chk2 MIM (Figure 3.5) shows that ATM phosphorylates substrate TP53BP1 (shown on the MIM as 53BP1). The G-test was applied to the subset of PNNL data [10] having TP53BP1 as a substrate. Those phospho-peptides that were found to be statically significant were retained as a query data set for input into my alignment code and aligned against the list of phospho-peptides generated from NetworKIN shown in Figure 3.7. This list contains phospho-peptide sequence EPVEQDSS*QPSLPLV (circled in Figure 3.7). This phospho-peptide sequence was also found to be a sequence that was significantly enhanced by exposure to 50cGy of radiation in the PNNL data [10]. Thus this particular phospho-peptide sequence found in PNNL data [10] can then be annotated with the information for this phospho-peptide that exists in the NetworKIN database.

All <input type="checkbox"/>	Substrate	Position	Context Score	Kinase	PhosphoPeptide
1 <input type="checkbox"/>	TP53BP1	S6	0.998	ATM	QLDSDFS
2 <input type="checkbox"/>	TP53BP1	S25	0.998	ATM	PCLIEDSPESQVL
3 <input type="checkbox"/>	TP53BP1	S29	0.998	ATM	IEDSQPEQVLEDD
4 <input type="checkbox"/>	TP53BP1	S105	0.998	ATM	LDTCGSIQVIEQLP
5 <input type="checkbox"/>	TP53BP1	S176	0.998	ATM	GFGVLELQSQDVEE
6 <input type="checkbox"/>	TP53BP1	S178	0.998	ATM	GVLELSQSQDVEENT
7 <input type="checkbox"/>	TP53BP1	T302	0.998	ATM	PEPEVLStQEDLFDQ
8 <input type="checkbox"/>	TP53BP1	S452	0.998	ATM	PGSLPIPSPQFSDH
9 <input type="checkbox"/>	TP53BP1	T543	0.998	ATM	IDEDGENTQIEDTEP
10 <input type="checkbox"/>	TP53BP1	S580	0.998	ATM	QDGEVQLQNDKTK
11 <input type="checkbox"/>	TP53BP1	S831	0.998	ATM	EPVEQDSQPSLPLV
12 <input type="checkbox"/>	TP53BP1	T855	0.998	ATM	QELQQPQtQEKTNS
13 <input type="checkbox"/>	TP53BP1	S892	0.998	ATM	GKPSAHAQSFCESS
14 <input type="checkbox"/>	TP53BP1	S1104	0.998	ATM	KDPVSPAQKMWIQG
15 <input type="checkbox"/>	TP53BP1	T1171	0.998	ATM	PETVSAAtQTIKNVC
16 <input type="checkbox"/>	TP53BP1	S1290	0.998	ATM	CETEVSPPQTGGSSG

Figure 3.7 NetworKIN results forTP53BP1

3.7 Data to Signaling Pathway

This entire *signaling pathways to data* approach described in section 3.6 is limited by the amount of data existing in MIMs and the number of substrates whose kinase are predicted in NetworKIN. So another method was tried to interpret more of the

statistically significant peptides in PNNL data [10]. This approach, which I call *data to signaling pathways* starts with the statistically significant phospho-peptides from PNNL data [10] and looks for information about them in various databases.

3.7.1 Submission to Phospho.ELM

After collecting the statistically significant phospho-peptides within PPD database, I export them to a text file and split the large file into smaller files for submission to Phospho.ELM [7]. From Phospho.ELM, I could then link to NetworkKIN [8]. The diagram of the method is shown in Figure 3.1. Phospho.ELM is one of several databases of experimentally observed phosphorylation sites. Version 7.0 of this database contains 4078 phosphorylated- protein sequences that include 12025 phosphoserine (S), 2362 threonine (T), and 2083 tyrosine (Y) phosphorylation sites. About 21% of the sites in the database are annotated with the kinase responsible for the phosphorylation.

I was able to download the database and import it into a local PPD database to use for data mining. Instead of writing a tool to align my peptides with the peptides in Phospho.ELM, I chose to develop a process to query Phospho.ELM and make use the search tool available from the Phospho.ELM website. Phospho.ELM website contains links to other related databases and a batch BLAST search tool, used for aligned peptides from user data with the peptides in Phospho.ELM. BLAST stands for Basic Local Alignment Search Tool.

To link to batch BLAST, I wrote a query to return a list of the peptides I wished to enter into BLAST. I exported the query results as a text file, Since Batch BLAST only allows for processing of 20 peptides at a time, I wrote a file splitter utility in Visual Basic

Script that would take a text file, a name for the split files (to be appended with the sequence of “_1, “_2, etc.), and a directory to write the split files (Appendix C).

From the Batch BLAST entry screen, I could browse to the location of the split files, select one and submit it for processing. Since there were more than 20 statistically significant peptides in PNNL data [10], it was necessary to split the data into smaller groups to be processed separately through BLAST.

Table 3.8 shows an example Batch BLAST file limited to 20 submissions. The format of the file is called, FASTA [59] in which submission must be in pairs of lines. The first line of the set starts with “>” followed by a description and the second line holds the sequence to be submitted.

Output from BLAST Phospho.ELM can be viewed in html format as well as raw text format. The html output in Figure 3.8 shows some of the additional information supplied by Phospho.ELM including links to related information contained in other databases such as NetworkKIN and PubMed. The results in Figure 3.8 are for the first submission at the top of Table 3.8.

> 1
WLNSGRGDEASEEGQNGSsPK
> 2
AsLGSLEGEAEAEASSPK
> 3
KGDRsPEPGQTWTR
> 4
SKGHYEVTGSDDEtGKLQGGVSLASK
> 5
TNGHsPSQPR
> 6
GINGGPSRMsPK
> 7
LAEDEGDsEPEAVGQSR
> 8
SSSSASPSsPSSR
> 9
SAsADNLTLP
> 10
LLGGTRtPINDAS
> 11
PFSAPKPQtSPSPKR
> 12
RLsYNTASNK
> 13
METEADAPsPAPSLGER
> 14
GLYDGPVCEVSVtPK
> 15 1
ELVEPLtPSGEAPNQALLR
> 16
sTPPPNNLVNPVQELETER
> 17
NVFEVGPGDsPTFPR
> 18
RTNsTGGSSGSSVGGGSGK
> 19
ALPAAAEDGsPVFGEGPPSLK
> 20
SAMDSPVPASMFAPePsSPGAAR

Table 3.8 Example Phosph.ELM BLAST input file

phosphoELM Currently 19650 instances in [Phospho.ELM](#) database.

All results in raw text format [here](#)

□ **Results for: 1 (1)**

>1 (1)
 WLNSGRGDEASEEGQNGSSPK

Note: The ranking of the alignments is according to the position on the query sequence

Matched Site (from Phospho.ELM)	Matched Site Substrate	Species	Position in the query sequence	Alignment	Kinase(s) upstream of matched site	PubMed Reference(s)
P35611_465_S	Alpha adducin	Homo sapiens	17**	Query: 14 GQNGSSPK 21 GQNGSSPK Sbjct: 1 GQNGSSPK 8	-	15302935 17081983

** please be aware that the specified position in the query sequence may not always be strictly correct due to partial alignment of the peptides and may not correspond to the phospho amino acid specified in the peptide name.

Figure 3.8 BLAST Output for 2-Rad-GTest -1.html

3.7.2 Combining Phospho.ELM with NetworKIN in PPD database

The html format is useful for viewing the results but I used the raw text format to load the results to the PPD database. This was done manually during my research: however, the process could be automated using a program outside the PPD database that sends database commands to the database or by developing a Visual Basic module inside the PPD database to load the files on user command.

Tools that I developed add value to the data mining process to create submission files to Phospho.ELM and then correlate and load the raw output format back into the PPD database for further processing. See Figure 3.2 for diagram for this method. As I developed my process I had to manually append the raw text files into a single file and then load the combined file into my PPD database. NetworKIN and Phospho.ELM

databases were also loaded into PPD database. With the BLAST output in the PPD database, I was able to directly query the extended information from NetworKIN and Phospho.ELM database related to the peptides returned from the tools for each database. The NetworKIN database (252 KB Gzipped file) contained 7143 records. PhosphoELM database (3542 KB Gzipped file) contained 9998 records.

3.7.3 Summarizing the Phospho.ELM + NetworKIN Output

By using the results from BLAST, I was able to merge together extended information found in the Phospho.ELM database and the NetworKIN database with PNNL data [10]. The results are presented in

Table 3.9 and

Table 3.10. While just a few fields from Phospho.ELM and NetworKIN were used to annotate PNNL data [10] in these tables, other fields are available. The relational database supporting the data store allows ad-hoc queries to merge, summarize, and report any fields that are found to be related through the BLAST output.

Both the motif score and the context score are used to assist in predicting the validity of the kinase family for a specific substrate. Scores range from 0 to 1 where the higher the score, the more likely the predicted kinase is valid. Substrates are listed multiple times in the summary tables because there were multiple statistically significant phospho-peptides associated with those substrates.

2RadPhosphoELM+Networkin				
substrate	position	kinase_family	context_score	motif_score
ATXN2L	S409	cdk5	0.936	0.586
ATXN2L	S409	cdk5	0.961	0.586
ATXN2L	S409	cdk5	0.962	0.586
ATXN2L	S409	cdk5	0.963	0.586
BAZ1B	S1468	CKII	0.973	0.66
BM28	S27	cdk5	0.968	0.708
BM28	S27	cdk5	0.989	0.708
BM28	S27	cdk5	0.994	0.708
BM28	S27	cdk5	0.996	0.708
BM28	S27	GSK3	0.989	0.515
BM28	S27	GSK3	0.991	0.515
BM28	S27	p38MAPK	0.978	0.531
BM28	S27	p38MAPK	0.988	0.531
BM28	S27	p38MAPK	0.991	0.531
BM28	S27	p38MAPK	0.992	0.531
BM28	S27	p38MAPK	0.993	0.531
BM28	S41	cdk5	0.968	0.557
BM28	S41	cdk5	0.989	0.557
BM28	S41	cdk5	0.994	0.557
BM28	S41	cdk5	0.996	0.557
CFP1	S326	PKA	0.416	0.811
CFP1	S326	PKA	0.459	0.811

Table 3.9 2RadPhosphoELM+Networkin Annotation Query Results

50RadPhosphoELM+Networkin				
substrate	position	kinase_family	context_score	motif_score
CFP1	S326	PKA	0.416	0.811
CFP1	S326	PKA	0.459	0.811
CFP1	S326	PKA	0.499	0.811
CFP1	S326	PKA	0.512	0.811
CFP1	S326	PKB	0.429	0.588
CFP1	S326	PKB	0.473	0.588
CFP1	S326	PKB	0.496	0.588
CFP1	S326	PKB	0.505	0.588
CFP1	S326	PKB	0.516	0.588
CFP1	S326	PKB	0.519	0.588
CFP1	S326	RSK	0.442	0.569
CFP1	S326	RSK	0.504	0.569
CFP1	S326	RSK	0.513	0.569
CFP1	S326	RSK	0.515	0.569
CFP1	S326	RSK	0.518	0.569
CFP1	S326	RSK	0.519	0.569
CHD3	S1601	CKI	0.911	0.505
CHD3	S1601	CKI	0.98	0.505
CHD3	S1601	CKI	0.986	0.505
CHD3	S1601	CKI	0.989	0.505
CHD3	S1601	GSK3	0.989	0.536

Table 3.10 5RadPhosphoELM+Networkin Annotation Query Results

CHAPTER 4 SUMMARY AND FUTURE WORK

Key observations from this work include:

1) Data mining's complexity increases with the number of samples or categories.

This was observed when comparing all of the PNNL data [10] to the NetworKIN results. It is extremely important to reduce the number of samples or categories to reduce the effort needed to apply data mining methods. Using tests of statistical significance as a filter can dramatically reduce the number of phospho-peptides by identifying those that are of serious interest prior to applying the methods of data mining,

2) Equivalence classes may be applied to proteomic data sets generated using mass spectroscopy. Such data sets contain varying fragments from the original sample. Equivalence classes may be defined to aid in forming collections of equivalent peptides. The abundance of individual fragments may be aggregated for the class of equivalent peptides. While the occurrences of a single peptide may have little or weak statistical significance, the aggregated occurrences of the equivalent peptide collection may give greater statistical power to the test of statistical significance. The process of identifying equivalence classes, aggregating collections of equivalent peptides, and performing statistical tests are greatly facilitated by automation tools. The tools developed as part of the research for this thesis have been demonstrated to have facilitated this first stage of data analysis, and

3) The workflow process is dependent on access to external data repositories. The access methods supported by repositories may enhance or inhibit access to the data contained in the repositories. The software associated with the Phospho.ELM database

was extremely helpful in enabling batch processing of lists of phospho-peptides and gaining access to additional data that was relevant to their biological functions.

Three general approaches were identified for data mining/interpretation of phospho-proteomics data in order to gain insight into signaling pathways. They are

- 1) Signaling Pathways to Data,
- 2) Data to Signaling Pathways, and
- 3) Bootstrap method of Puente, et al. [26].

The work in this thesis investigated the first two of these three approaches:

Signaling Pathways to Data, and Data to Signaling Pathways. The third approach builds a kinase-substrate network by using existing databases that contain this information. The network is then visualized and analyzed. This approach could be implemented in the future for the phospho-peptide data set used in this research.

A core component of my tools is a relational database, which may not be familiar to the end user of these tools. The tools and processes described in the thesis by Guo [32] could be incorporated into my tools to allow a user to add additional databases and queries to the PPD database for further data annotation. The process and tools describe in this thesis could be encapsulated along with Guo's techniques into a packaged toolkit with a graphical user interface that further simplifies the process and access to the tools for end users.

BIBLIOGRAPHY

- [1] G. Manning, D. B Whyte, R. Martinez, T. Hunter and S. Sudarsanam, “The protein kinase complement of the human genome”, *Science*, vol. 298, pp 1912-1934, 2002.
- [2] P. Cohen, “The origins of protein phosphorylation”, *Nat. Cell Biol.*, vol. 4, pp E127-E130, 2002.
- [3] S. Beausoleil, M. Jedrychowski, D. Schwartz et al., “Large-scale characterization of Hela cell nuclear phosphoproteins”, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 101, pp 12130-12135, 2004.
- [4] J. Olsen, B. Blagoev, F. Gnad, et al., “Global, in vivo, and site-specific phosphorylation dynamics in signaling networks”, *Cell*, vol. 127, pp 635-648, 2006.
- [5] M. Hjerrild, A. Stensballe, T. E. Rasmussen, et al., “Identification of phosphorylation sites in protein kinase A substrates using artificial neural networks and mass spectrometry”, *J. Proteome Res.*, vol. 3, pp 426-433, 2004.
- [6] R. Linding, L. J. Jensen, G. J. Osteimer, et al., “Systematic Discovery of in vivo phosphorylation networks”. *Cell*, vol. 129, pp 1415-1426, 2007.
- [7] F. Diella, C. M. Gould, C. Chica, A. Via and T. J. Gibson., “Phospho.ELM: a database of phosphorylation sites – update 2008”, *Nucleic Acids Res.* vol. 36, pp D240-D244, 2008.
- [8] R. Linding, L. J. Jensen, A. Pasculescu et al., “NetworKIN: a resource for exploring cellular phosphorylation networks”, *Nucleic Acids Res*, vol. 36, pp D695-D699, 2007.
- [9] F. Yang, S. Wu, D. L. Stenoien, et al., “Combined pulsed-Q dissociation and electron transfer dissociation for identification and quantification of iTRAQ-labeled phosphopeptides”, *Anal. Chem.*, in press.
- [10] F. Yang, L. Pasa-Tolic, M. A. Gristenko, et al., “Phosphoproteomic profiling of low dose radiation responses in a skin model system”, *Abstracts of the Low Dose Radiation Research Program Investigators Workshop VIII*, p 127, Washington D. C., April 6-8, 2009.
- [11] R. Sokal and F. Rohlf, *Biometry: the Principles and Practice of Statistics in Biological Research*, New York: Freeman, 1994.

- [12] J. W. H. Wong, M. J. Sullivan and G. Coney, "Computational methods for the comparative quantification of proteins in label-free LCn-MS experiments", Briefings in Bioinformatics, vol. 9, pp 156-165, 2007.
- [13] B. Zhang, N. VerBerkmoes, M. Langston, et al., "Detecting differential and correlated protein expression in label-free shotgun proteomics", J. Proteome Res., vol. 5, pp 2909-2918, 2006.
- [14] W. Old, K. Meyer-Arendt, L. Aveline-Wolf, et al., "Comparison of label-free methods for quantifying human proteins by shotgun proteomics. Mol. Cell. Proteomics, vol. 4, pp 1487-1502, 2005.
- [15] T. Beissbarth, L. Hyde, G. K. Smyth, et al., "Statistical modeling of sequencing errors in SAGE libraries", Bioinformatics, vol. 20, pp i31-i39, 2004.
- [16] Y. Pommier, J. N. Weinstein, M. I. Aladjem and K. W. Kohn, "Chk2 molecular interaction map and rationale for Chk2 inhibitors", Clin. Cancer Res., vol. 12, pp 2657-2661, 2006.
- [17] C. von Mering, L. J. Jensen, B. Snel, et al., "STRING: known and predicted protein-protein associations, integrated and transferred across organisms", Nucleic Acids Res., vol. 33, pp D433-D437, 2005.
- [18] Rashmi Raj Biological Data Mining Stanford-CS374 lecture notes, 10/31/1996
- [19] John L. Pfaltz Christopher M. Taylor, "Closed Set Mining of Biological Data", BIODDD02: Workshop on Data Mining in Bioinformatics (with SIGKDD02 Conference), 2002
- [20] Sanghamitra Bandyopadhyaya, et al. Text - Analysis of Biological Data A software computing approach, September 2007
- [21] L Li, et al. Data Mining Techniques for cancer detection using serum proteomic profiling, Artif Intell Med, vol. 32, no.2, pp.71- 83. Oct 2004
- [22] Usama Fayyad, et al, From Data Mining to Knowledge Discovery, Ai Magazine, vol. 17, no. 3, pp. 37-54, 1996
- [23] D Devos, et al., Practical limits of Function Prediction, Proteins vol. 41, no.1, pp 98-107, 2000
- [24] Jacquelyn S Fetrow, et al., Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to Glutaredoxins/Thioredoxins and T₁Ribonucleases, Journal of Molecular Medicine, , vol. 281, , Issue 5, pp. 949-968, September 1998

- [25] C Orengo, et al., Text Bioinformatics, Chapter 5- Function Prediction from protein sequences, ISBN: 978-1-85996-054-7
- [26] Lawrence G. Puente, et al. Reconstructing the Regulatory Kinase Pathways of Myogenesis from Phosphopeptide Data, *Molecular & Cellular Proteomics* vol. 5, pp.2244-2251, 2006
- [27] M Marcantonio, et al., Combined Enzymatic and Data Mining Approaches for Comprehensive Phosphoproteome Analysis, *Molecular & Cellular Proteomics* vol. 7, pp. 645-650, 2008
- [28] Huang, P. H., et al., Uncovering therapeutic targets for glioblastoma: a systems biology approach, *Cell Cycle*, vol. 6, pp. 2750-2754, 2007
- [29] B Bodenmiller, et al., PhosphoPep—a phosphoproteome resource for systems biology research in *Drosophila* Kc167 cells, *Mol Syst Biol.* 2007, vol. 3, pp.139, 2007
- [30] N Tedford, et al., Illuminating signaling networks functional biology through quantitative phosphoproteomic mass spectrometry, *Functional Genomics and Proteomics*, Online October 2008
- [31] G Tzanis, Departement of Informatics Aristotle University of Thessaloniki, Greece, Biological Data Mining,
- [32] X Guo, Thesis “Exploring Automation of Safe Database Schema Evolution for Non-Skilled Users” Washington State University, 2002
- [33] Marjo deGraauw, Text - Phospho-Proteomics Methods and Protocols,2009
- [34] G Tzanis, A Novel Approach for the Accurate Prediction of Translation Initiation Sites, *Lecture Notes in Computer Science*, vol. 4345, pp.92-103, 2006
- [35] B Bodenmiller, et al., PhosphoPep—a phosphoproteome resource for systems biology research in *Drosophila* Kc167 cells, *Mol Syst Biol.* 2007, vol. 3, pp.139, 2007
- [36] Z Du Improvement of the Needleman-Wunsch Algorithm, *Lecture Notes in Computer Science*, vol. 3066, pp.792-797, 2004
- [37] Henikoff, S.; Henikoff, J.G. (1992). Amino Acid Substitution Matrices from Protein Blocks, *Proc. Natl. Acad. Sci. USA*, vol. 89, no. 22, pp. 10915–10919, 1992
- [38] Altschul, S. F., et al., OPTIMAL SEQUENCE ALIGNMENT USING AFFINE GAP COSTS, *Bulletin of Mathematical Biology*, vol. 48, no.5-6, pp 603-616, 1986

- [39] Whishart, D.S. (2002). Tools for Protein Technologies. In Sensen, C.W. (Ed.), *Biotechnology*, vol. 5b, Genomics and Bioinformatics, pp.325-344, Wiley-VCH.
- [40] M Cannataro, et al., Using Proteus for Modeling Data Analysis of Proteomics Experiments on the GRID, *Lecture Notes in Computer Science*, vol. 3292, pp. 232-243, 2004
- [41] Nakayasu, et al., Phosphoproteomic analysis of the human pathogen *Trypanosoma cruzi* at the epimastigote stage, *Proteomics*, vol. 9, no.13, pp.3489-3506, July 2009
- [42] Macek, et al. Global and site-specific quantitative phosphoproteomics: principles and applications, *Annual review of pharmacology and toxicology*, vol. 49, pp. 199-221, October 2009
- [43] Zanivan, et al. Solid Tumor Proteome and Phosphoproteome analysis by High Resolution Mass Spectrometry *J. Proteome Res.*, vol. 7, no.12, pp 5314–5326, 2008
- [44] Y Lim, et al., Mining the Tumor Phosphoproteome for cancer markers, *Clinical cancer research*, vol. 11, pp.3163-3169, May 2005
- [45] Macek, et al. Global and site-specific quantitative phosphoproteomics: principles and applications, *Annual review of pharmacology and toxicology*, vol. 49, pp. 199-221, October 2009
- [46] I Ahmad, et al., Phosphoproteome sequence analysis and significance: Mining association patterns around phosphorylation sites utilizing MAPRes, *Journal of Cellular Biology*, vol. 108, no. 1, pp. 64-74, June 2009
- [47] F Gnad, High-accuracy identification and bioinformatic analysis of *in vivo* protein phosphorylation sites in yeast, vol. 9, no. 20, pp.4642 – 4652, September 2009.
- [48] F. Cerqueira, et al., Improving Phosphopeptide/protein Identification using a New Data Mining Framework for MS/MS Spectra Preprocessing, *OMIC Online*, March 2009
- [49] H Xu, et al., A hierarchical MS²/MS³ database search algorithm for automated analysis of phosphopeptide tandem mass spectra, *Proteomics*, vol. 9 no. 7, pp. 1763– 1770, February 2009
- [50] R Higdon, et al., A Predictive model for identifying proteins by a single peptide match, *Bioinformatics* 2007, vol. 23, no.3, pp.277-280, 2007

- [51] H Xu, et al., MassMatrix: A database search program for rapid characterization of proteins and peptides from tandem mass spectrometry data, *Proteomics*, vol. 9 no. 6, pp. 1548 – 1555, February 2009
- [52] BD Halligan, DeNovoID: a web-based tool for identifying peptides from sequences and mass tags deduced from de novo peptide sequencing by mass spectrometry, *Nucleic Acids Research* 2005 vol. 33(Web Server Issue),W376-W381
- [53] Z Zhang, et al., A novel scoring schema for peptide identification by searching protein databases, *BMC Bioinformatics*, vol. 7, pp.222, 2006
- [54] P Shannon et al., The Gaggle: An open-source software system for Integrating bioinformatics software and data sources, *BMC Bioinformatics* 2006, vol. 7, pp. 176, 2006
- [55] Desiere, et al., Integration with the human genome of peptide sequences obtained by high-throughput mass spectrometry. *Genome Biol.*, vol. 6, no. 1, Published online 2005
- [56] Yates, et al. Mining Genomes: Correlating Tandem Mass Spectra of Modified and Unmodified Peptides to Sequences in Nucleotide Databases *Anal. Chem.*, vol. 67, no.18, pp 3202–3210,1995
- [57] Obenaus, et al., Scansite 2.0: proteome-wide prediction of cell signaling interactions using short sequence, *Nucleic acids research*, vol. 31, no.13, pp. 3635-3641, 2003
- [58] Olsen, et al. Global, in vivo, and site-specific phosphorylation dynamics in signaling networks, *Cell* 2006 Nov 3, vol 127, no.3, pp.635-48, 2006 Kohn et al., Molecular interaction maps of bioregulatory networks: a general rubric for systems biology, *Molecular Biology of the cell*, vol. 17,no. 1, pp 1-13 2006.
- [59] <http://www.ncbi.nlm.nih.gov/blast/fasta.shtml>
- [60] <http://networkin.info/search.php>.
- [61] <http://discover.nci.nih.gov/mim/index.jsp>.
- [62] Quinlan J.R. , *Text C4.5: Machine Learning Programs*: Morgan Kaufmann, 1993
- [63] <http://www.uniprot.org/>
- [64] <http://www.expasy.ch/prosite/>
- [65] http://searchsqlserver.techtarget.com/sDefinition/0,,sid87_gci212885,00.html

[66] McDonald, J.H. 2009. Handbook of Biological Statistics, 2nd ed. Sparky House Publishing, Baltimore, Maryland.

[67] <http://www.avatar.se/molbioinfo2001/dynprog/dynamic.html>

APPENDIX A NEEDLEMAN-WUNSCH IMPLEMENTATION

```
#!/usr/bin/perl -I . -I ../perllib
#####
### Copyright (c) 2002 Rex A. Dwyer.
### Distributed with the book
### Genomic Perl: From Bioinformatics Basics to Working Code
### copyright (c) 2002 Cambridge University Press
### under terms described therein.
#####
##package SeqReader;

##use strict;
use Util;
##use SeqReader;
use SimpleFormatReader;

#####
## An implementation of the Needleman-Wunsch algorithm
## for global alignment of DNA sequences.
#####

#####
## GLOBAL VARIABLES -
## Modified to add affine penalty instead of -2 for gap
#####
my (@Mbb,@Mb_,@M_b); ##alignment matrix; fill by similarity
my $g = -1; ## gap penalty.
my $h = -1;
```

```
my %lod; ## matrix of lod-scores for residue pairs is store here.
```

```
#####
```

```
sub fillLod {
```

```
## Reads a scoring matrix from the specified file; stores it in %lod.
```

```
## RETURNS: nothing.
```

```
#####
```

```
    my ($matrixFile) = @_; ## name of file from which to read matrix.
```

```
    open LOD, $matrixFile;
```

```
##print ($matrixFile "\n");
```

```
    my ($trash,@residues) = split /\s+/, <LOD>;
```

```
    while (<LOD>) {
```

```
        my ($r,@scores) = split;
```

```
        foreach (0..19) {
```

```
            $lod{$r.$residues[$_]} = $scores[$_];
```

```
##        print($lod{$r.$residues[$_]},"\n");
```

```
##        print($r.$residues[$_]},"\n");
```

```
        }
```

```
    }
```

```
}
```

```
##    print($lod{A.1},"\n");
```

```
my @residues = split //,"acdefghiklmnpqrstvw"; ## list of 20 amino acids
```

```
#####
```

```
sub p {
```

```
## given two amino acids or nucleotides, compares
```

```
## them and returns a match reward (+1) or mismatch
```



```

## penalty (-1). For amino acids, it should normally
## be replaced with a more complicated function.
## RETURNS: numerical reward/penalty.
#####
#####
## Change this sub to lookup score in scoring matrix instead.
#####
    my ($aa1, $aa2) = @_ ; ## residues/bases to be compared.
## print ($aa1,$aa2, "\n");
## return ($aa1 eq $aa2)?1:-1;
## my ($query, ## query sequence (string)
##     $qPos, ## position of hit in query sequence
##     $qId, ## identifier (database label) of query
##     $target, ## target sequence (string)
##     $tPos, ## position of hit in target sequence
##     $tId, ## identifier (database label) of query
    my($matchscore); ##score from scoring matrix

## my @target = ("-", split //,$target);
## my @query = ("-", split //,$query);
## my ($lo,$hi) = (0,2);

    my $matchscore = $lod{$aa1.$aa2};
##print ($matchscore,"\n");
    return ($matchscore);

}

#####
sub max {
## Given any positive number of numerical arguments,

```

```

## RETURNS: the largest.
#####
my ($m,@l) = @_; ## numerical values.
foreach my $x (@l) { $m = $x if ($x > $m); }
return $m;
}

#####
sub similarity {
## Determines score of best alignment of strings $s and $t
## by filling in alignment matrixes @Mbb,@Mb_, and @M_b.
## RETURNS: nothing; fills @M.
#####
my($s,$t) = @_; ## sequences to be aligned.
## foreach my $i (0..length($s)) { $M[$i][0] = $g * $i; }
## foreach my $j (0..length($t)) { $M[0][$j] = $g * $j; }
foreach my $i (0..length($s)) { $Mbb[$i][0] = $g*$i;
                                $Mb_[$i][0] = $g*$i;
                                $M_b[$i][0] = $g*$i;}
foreach my $j (0..length($t)) { $Mbb[0][$j] = $g*$j;
                                $Mb_[0][$j] = $g*$j;
                                $M_b[0][$j] = $g*$j;}
foreach my $i (1..length($s)) {
  foreach my $j (1..length($t)) {
    my $p = p(substr($s,$i-1,1),substr($t,$j-1,1));
    $Mbb[$i][$j] =
    max($Mbb[$i-1][$j-1] + $p,
        $Mb_[$i-1][$j-1] + $p,
        $M_b[$i-1][$j-1] + $p);
    $M_b[$i][$j] =
    max($M_b[$i][$j-1] + $g,

```

```

        $Mbb[$i][$j-1] + $h + $g,
        $Mb_[$i][$j-1] + $h + $g);
$Mb_[$i][$j] =
max($Mb_[$i-1][$j] + $g,
    $Mbb[$i-1][$j] + $h + $g,
    $M_b[$i-1][$j] + $h + $g);

##print($M[$i][$j], "\n");

    }
}

return
(max($Mbb[length($s)][length($t)], $M_b[length($s)][length($t)], $Mb_[length($s)][length($t)] );
}

#####
sub getAlignment {
## Reconstructs best alignment of strings $s and $t using information
## stored in alignment matrix @M by similarity. Recursive.
## RETURNS: list of two strings representing best alignments.
## These strings are $s and $t with gap symbols inserted.
#####

my ($s,$t) = @_ ; ## sequences to be aligned.
my $simscore;
my $simscore = similarity($s,$t);

my ($i,$j) = (length($s), length($t));
return ( "-"x$j, $t) if ($i==0);
return ( $s, "-"x$i) if ($j==0);
## return [$simscore, "-"x$j, $t] if ($i==0);

```

```

## return [$simscore, $s, "-"x$i] if ($j==0);

my ($sLast,$tLast) = (substr($s,-1),substr($t,-1));

if ($Mbb[$i][$j] == max($Mbb[$i][$j],$M_b[$i][$j],$Mb_[$i][$j])) {
  ## Case 1
  ## last letters are paired in the best alignment
  my ($sa, $ta) = getAlignment(substr($s,0,-1), substr($t,0,-1));
  return ($sa . $sLast , $ta . $tLast );
## return [$simscore, $sa . $sLast , $ta . $tLast ];

} elsif ($Mb_[$i][$j] == max($Mbb[$i][$j],$M_b[$i][$j],$Mb_[$i][$j])) { ## Case 2
  ## last letter of the first string is paired with a gap
  my ($sa, $ta) = getAlignment(substr($s,0,-1), $t);
  return ($sa . $sLast , $ta . "-");
## return [$simscore, $sa . $sLast , $ta . "-"];

} else { ## Case 3: last letter of the 2nd string
## is paired with a gap
  my ($sa, $ta) = getAlignment($s, substr($t,0,-1));
  return ($sa . "-" , $ta . $tLast );
## return [$simscore, $sa . "-" , $ta . $tLast ];
}
}

#####
sub findExactMatches {
## Finds target string(s) $t with exact alignment to query
## string $s. If exact match is found searching stops else
## get alignment is called to compare and score $s with all $t to
## produce top 10 alignments.

```

```

## RETURNS: list of two strings representing best alignments.
## These strings are $s and $t with gap symbols only inserted.
#####
my ($s,$t) = @_; ## sequences to be aligned.
my $mscore;
my $mscore = 1;
my $sphosphosite = 0;
my $tphosphosite = 0;
my $match = 1;
my $i = 0;
my $leftendoftargetreached = 0;
my $rightendofqueryreached = 0;
my $leftendofqueryreached = 0;
my $rightendoftargetreached = 0;
my $stringsoverlap = 0;
my $sllength = 0;

if (length($s) > length($t)) {
    $sllength = length($t);
} else {
    $sllength = length($s);
}

## print "SL $sllength\n";

## ($i,$j) = (length($s), length($t));
    $sphosphosite = index($s,"*");
##print "ssite",$sphosphosite;
    $tphosphosite = index($t,"*");
##print "tsite",$tphosphosite;

```

```

my $tloc = $tphosphosite;

## while ($match > 0) {
  for ($i = $sphosphosite - 1; $i >= 0; $i--) {
##   print "$i";
    if ($match > 0){
      $sl = substr($s,$i,1);
      if ($tloc > 0){
        $tloc = $tloc - 1;
        $tl = substr($t,$tloc,1);
        if ($sl eq $tl){
##         print $sl,"\n";
##         print $tl,"\n";
          $mscore = $mscore + 1;
          } ##sl=tl
        else {$match = 0;
          }

        }##else {$match = 0;
##   }
}##end if $match
} ##end for

##   print "TLOC1 $tloc\n";
##   print "Match1 $match\n";

if (($tloc == 0)and ($match == 1)){
  $leftendoftargetreached = 1;
}

##   print "LET $leftendoftargetreached\n";
if (($tloc > 0) and ($match == 1)){
  $leftendofqueryreached = 1;
}

```

```

    }
##   print "LEQ $leftendofqueryreached\n";
    $tloc = $tphosphosite;
##   $match = 1;

    for ($i = $sphosphosite + 1; $i <=length($s)-1; $i++) {
        if ($match > 0){
            $sl = substr($s,$i,1);
##print $sl,"\n";

##       if ($tloc < (length($t))){

            if ($tloc < (length($t)-1)){
                $tloc = $tloc + 1;
                $tl = substr($t,$tloc,1);
##       print $tl,"\n";

                if ($sl eq $tl) {
##       print $sl;
##       print $tl,"\n";

                    $mscore = $mscore + 1;
                }else { $match = 0
                }
            }##else { $match = 0;
            #}
        }##end if $match
    } ##end for

##   print "TLOC2 $tloc\n";
##   print "Match2 $match\n";

```

```

##   if (($tloc > 0)and ($match == 1)){
      if (($tloc ==(length($t)-1))and ($match == 1)){
          $rightendoftargetreached = 1;
      }
##   print "RET $rightendoftargetreached \n";

      if (($tloc < length($t)) and ($match == 1)){
          $rightendofqueryreached = 1;
      }
##   print "REQ $rightendofqueryreached \n";

if (($leftendoftargetreached == 1) and ($rightendofqueryreached == 1)){
    $stringsoverlap = 1;
}
if (($leftendofqueryreached == 1) and ($rightendoftargetreached ==1)){
    $stringsoverlap = 1;
}
##   print "SO $stringsoverlap \n";

    if ($stringsoverlap == 1){
        $mscore = $sllength;
##       print "Score $mscore\n";

    }

## } ## end while
##   print "returning score $mscore\n";
    return ($mscore);
}

#####

```



```

## MAIN PROGRAM
#####
{
#####
## Need call to sub that reads input files(output from Networkin and
## David's peptides) and fills arrays. Then add two for loop to this
## main program to search for every networkin result in PNNL data.
## Modify to write output to a file
#####
## Read in scoring matrix specified on command line -- default blosum62.
##fillLod($ARGV[2] || "blosum62");
fillLod("blosum62plus");

my $qsite = 0;
my $tsite = 0;
my $x = 0;

my($s,$t) = @_ ; ## sequences to be aligned.

## my ($s,$t) = ("SASKATCHEWAN", "SESQUICENTENNIAL");
## my ($s,$t) = ("S*QLDSDFS", "TGSSSPGGPPKPGS*QLDSMLGSLQSDLNK");
## my ($s,$t) = ("sldsdfs", "tgssspggppkpgsldsmlgslqsdlnk");

my $queryfile = SimpleFormatReader->new($ARGV[0]);
##my $queryfile = new SeqReader "queryfile";
##my $queryfile = SeqReader->new($ARGV[0]);

## print $queryfile,"\n";
## my $targetfile = SimpleFormatReader->new($ARGV[1]);
## my $targetfile = SeqReader->new($ARGV[1]);
## print $targetfile,"\n";

```

```

##241
  ## open (OUT,">BestAlignments");
  while (my $s = $queryfile->readSeq()) {
    ##for each query from NetworKin
    ##init array of bestAlignments to empty for each query $s in queryfile
    ## print OUT "Query String $s\n";
  ##   print "Query String $s\n";
    my @bestAlignments = ();
    my $matchfound = 0;
  ##   print "matchfound",$matchfound;
    my $targetfile = SimpleFormatReader->new($ARGV[1]);
  ##   while ((my $t = $targetfile->readSeq()) and ($matchfound = 0)){
      while (my $t = $targetfile->readSeq()) {
  ##     print "Target String $t\n";

##Add exact match code check
    my $exactmatchscore = 0;
    my $exactmatchscore = findExactMatches($s,$t);
    my $sllength = 0;
    if (length($s) > length($t)) {
      $sllength = length($t)
    }else {
      $sllength = length($s)}
##Definition of exact match.
##Match also if end of query string is reached one way and target ##string the other way
when comparing and match was not set to
##(mismatch)
    if ($exactmatchscore == $sllength) {
      $qsite = index($s,"*");
      $tsite = index($t,"*");
  ##print "qsite",$qsite,"\n";
  ##print "tsite",$tsite,"\n";

```

```

print "Query String ";

if ($qsite == $tsite){
    print " ";
}

if ($qsite < $tsite){
    for ($x = 0; $x <= ($tsite - $qsite); $x++) {
        print " ";
    }
}
##    print "Query String $s\n";
print "$s\n";

print "Target String";

if ($qsite > $tsite){
    for ($x = 1; $x < ($qsite - $tsite); $x++) {
        print " ";
    }
}
print " $t\n";
##    print "Target String $t\n";
##    print "matchscore", $exactmatchscore;
##    print "length", $sllength;
print "Exact match\n\n";
$matchfound = 1;
}else
{

```

```

## print "Query String $s\n";
## print "No exact match exists\n\n";
my $alignment = [similarity($s,$t), getAlignment($s,$t)];

    ### Add new alignment to sorted list.
### Need to save score, aligned
##string $s and $t

push @bestAlignments,$alignment;
if ($#bestAlignments > 9) {
    @bestAlignments =
        sort { $$b[0] <=> $$a[0] } @bestAlignments;
    ### Cut off list at 10 alignments.
    $#bestAlignments = min($#bestAlignments,9);
}
} ##end if matchfound
} ##end inner while loop for target queries

if ($matchfound == 0) {
    print "Query String $s\n";
    print "No exact match exists\n\n";

###write 10 best alignments for each queryfile query $s to a file ##instead of the screen
foreach (@bestAlignments) {
    ##print OUT (join("\n", @$_), "\n\n");
    print (join("\n", @$_), "\n\n");
    ##print(@$_, "\n\n");
} ##foreach
} ##exactmatchscore if

```

```
$targetfile->close();  
  } ##end outer while loop for queryfile queries  
$queryfile->close();  
## close(OUT);  
} ##MAIN
```

APPENDIX B GROUPS BY PHOSPHORYLATION SITE MATCH

E		F		G	H	I	J	K			L	M	N
Peptide	min_of_reference	control	2	cGy	count	50 cGy	Total	Description	Entrez	GeneName	Swiss	Prot	
1		count	Count	Count	Count	count	spectral		GeneID		Prot		
2	FGT*FGGLGSK	IP:IP00021812.1	2	2	2	2	4	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
3	FKAEPLPS*PK	IP:IP00021812.1	3	2	2	2	7	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
4	GDLS*Y*KAS*LGSLGEAEAEASSPK	IP:IP00021812.1	1	1	1	1	2	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
5	GDLS*Y*KASLGS*LEGEAEAEASSPK	IP:IP00021812.1	1	1	1	1	3	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
6	GDLS*Y*KASLGS*LEGEAEAEAS*SPK	IP:IP00021812.1	1				1	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
7	GGVT*GSPPEAS*TSKSGDGLK	IP:IP00021812.1				1	1	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
8	GGVT*GSPPEASISGSKGDLK	IP:IP00021812.1	1				1	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
9	GKGGVT*GSPPEASISGSKGDLK	IP:IP00021812.1				1	1	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
10	GKGGVT*GSPPEASISGSK	IP:IP00021812.1	5	4	2	11	11	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
11	GKGGVT*PEAS*TSKSGDGLK	IP:IP00021812.1				4	3	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
12	GKGGVT*PEAS*TSKSGDGLK	IP:IP00021812.1				1	1	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
13	GGVTGS*PEASIS*GSKGDLK	IP:IP00021812.1	3	1	3	7	7	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
14	GKGGVTGS*PEASIS*GSKGDLK	IP:IP00021812.1	2	3	2	7	7	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
15	GGVTGS*PEASISGSK*GDLK	IP:IP00021812.1	1	2	1	2	3	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
16	GKGGVTGS*PEASISGSK*GDLK	IP:IP00021812.1	1	3	1	5	5	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
17	GGVTGS*PEASISGSK	IP:IP00021812.1	5	8	8	21	21	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
18	GGVTGS*PEASISGSKGDLK	IP:IP00021812.1	7	9	7	23	23	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
19	GKGGVTGS*PEASISGSK	IP:IP00021812.1	7	10	9	26	26	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		
20	GKGGVTGS*PEASISGSKGDLK	IP:IP00021812.1	6	10	3	19	19	NEUROBLAST DIFFERENTIATION-ASSOCIATED PROTEIN AHNAK (FRAGMENT)	79026	AHNAK	Q09666		

APPENDIX C FILE SPLITTER UTILITY

The following script was contained in a file called FileSplit.vbs. It was executed from a command prompt within Windows.

In the following example command, I will assume:

- the FileSplit.vbs file is in the directory D:\BlastFiles.
- The file exported from PPD database is called BlastIn.txt.
- The split files are to have a base or leading name of BlastInput.

The command would then be:

```
D:\BlastFiles>wscript.exe FileSplit.vbs BlastIn.txt BlastInput D:\BlastFiles
```

Using wscript.exe provides a GUI for the program execution. As the program executes, information about the state of the program are supplied to the user. Possible messages include:

- input file does not exist...
- input file exists...
- file being split...
- input file not split...
- input file split...
- finished!

The resulting split files would be located in the D:\BlastInput directory and named BlastInput_1.txt, BlastInput_2.txt, and so on.

```

if WScript.Arguments.Count > 0 then
' retrieve the command line arguments
sourceFile = WScript.arguments.Item(0)
    basename = WScript.arguments.Item(1)
    targetDir = WScript.arguments.Item(2)
set objFSO = CreateObject("Scripting.FileSystemObject")
' check the file exists
    if objFSO.FileExists( sourceFile ) then
WScript.Echo ("input file exists...")
set fs = CreateObject("Scripting.FileSystemObject")
isNotSplit = True
TargetCount = 1
Step = 20
FileCount = 0
with fs.OpenTextFile(sourceFile)
    while Not .AtEndOfStream
        TargetCount = TargetCount + Step
        FileCount = FileCount + 1
' Create new file name
        newfile1 = targetDir + "\" + basename + "_" +
CStr(FileCount) + "." + objFSO.GetExtensionName(sourceFile)
set w1 = objFSO.CreateTextFile(newfile1, True)
While ((Not .AtEndOfStream) and (.Line < TargetCount))
    w1.WriteLine ">" + CStr(.Line)
    w1.WriteLine .ReadLine
WEND
w1.Close
if(( isNotSplit ) AND (Not .AtEndOfStream)) then
    isNotSplit = False
    WScript.Echo ("file being split...")
end if
wend
.Close
end with

```



```
if( isNotSplit ) then

    WScript.Echo ("input file not split...")
else
    WScript.Echo ("input file split...")
end if
else
    WScript.Echo ("input file does not exist...")
end if
WScript.Echo ("finished!")
end if
```

APPENDIX D STATISTICALLY SIGNIFICANT PEPTIDE LISTS

seq_top	2cGy GTestF=1.25
KLEKEEEEGISQES*S*EEEEQ	21.00957174
IVRGDQPAASGDS*DDDEPPPLPR	17.28844475
KLEKEEEEGIS*QES*S*EEEEQ	17.22096031
VADAKGDS*ES*EEDEDLEVPVPSR	16.96582788
ALPAAAEDGS*PVFGEGPPSLK	13.15988832
CHS*LGYNFIHK	11.18127519
HIKEEPLS*EEEPCTSTAIASPEKK	11.18127519
TCS*FGGFDLTNR	11.18127519
TNS*MQQLEQWIK	11.18127519
AITSLGSGS*PK	9.892638744
LAEDGDS*EPEAVGQSR	9.099177857
GS*PEEELPLPAFEK	8.614584457
LQALKEEPQTVPEMPGET*PPLS*PIDMESQER	8.614584457
WLAES*PVGLPPEEEDKLTR	8.614584457
WLNSGRGDEASEEGQNGSS*PK	8.585442247
ESPRPLQLPGAEGPAIS*DGEEGGGEPGAGGGAAGAAGAGR	7.768889745
ISYIPDEEVSSPS*PPQR	7.768889745
QQPPLGPSSSLLS*LPGLK	7.768889745
SAMDSPVPASMFPEPS*SPGAAR	7.768889745
SRAS*PATHR	7.768889745
SVSEINS*DDELSGK	7.768889745
VVDYSQFQES*DDADEDYGR	7.768889745
NYQQNYQNS*ESGEKNEGSESAPEGQAQQR	7.462564871
AQS*PTPSLPASWK	7.34968835
MLAES*DES*GDEESVSQTDKTELQNTLR	7.34968835
RVS*VCAETYNPDEEEEDTDPR	7.34968835
TLHCEGTEINS*DDEQESKEVEETATAK	7.34968835
VEMGTSSQNDVMSWIPQETLNQINKAS*PR	7.34968835
VPPPRS*PQAQEAPVNIDEGLTGCTIQLLPAQDK	7.34968835
DTSPDKGELVS*DEEEDT	6.956644583
SFS*LDPLMER	6.956644583
TNGHS*PSQPR	6.79320623
APQTSSS*PPPVR	6.45557352
QIASQFPPPTPPAMESQPLKVPANVAPQS*PPAVK	6.45557352
RNS*LTGEEGQLAR	6.45557352
SAS*ADNLTLP	6.45557352
SSTLSSSSMSLS*PK	6.45557352
VPAS*PLPGLER	6.45557352
VTTEIQLPSQS*PVVEEQSPASLSSLR	6.45557352
AS*LGSLEGEAEAEASSPK	6.416084556
DWEDDS*DEDMSNFDR	6.36425242
KGDRS*PEPGQWTR	6.36425242
RRT*PT*PPPR	6.107738608

ALS*IESLSPTDSSNGVNWR	6.10159956
GLSAS*LPDLSENWIEVK	6.10159956
GLYDGPVCEVSVT*PK	6.10159956
GSPVSEIGWETPPPE*PR	6.10159956
IGELGAPEVWGLS*PK	6.10159956
KLGDVSPT*QIDVSQFGSFKEDTK	6.10159956
KLS*NPDIFSSTGK	6.10159956
PFSAPKPQT*SPSPKR	6.10159956
RDS*LGAYASQDANEQQDLGKR	6.10159956
S*TPPPNNLVNPNVQELETER	6.10159956
SSLS*GDEEDELFGATLK	6.10159956
EGMNPSYDEYADS*DEDQHDAYLER	6.077118527
KSLPAPVAQRPD*PGGGLQAPGQK	6.077118527
ALSSGGSITS*PPLSPALPK	5.814792541
SKGHYEVTGSDD*GKLGSGVSLASK	5.814792541
KPNAGGS*PAPVR	5.295572855
KVVDYSQFQES*DDADEDYGR	5.295572855
AERS*PNQGK	5.164674396
ELVEPLT*PSGEAPNQALLR	5.164674396
GPPDFS*S*DEEREPT*PVLGSGAAAAGR	5.164674396
GQGSS*PVAMQK	5.164674396
HS*PTPQQSNR	5.164674396
HVAYGGYST*PEDR	5.164674396
IEVLPVDTGAGGYSGNSGS*PK	5.164674396
KDS*QNSSQHSVSSHR	5.164674396
LLGGTRT*PINDAS	5.164674396
MQFS*FEGPEK	5.164674396
MSDS*LDTDPSMLGSSK	5.164674396
NHS*GS*RT*PPVALNSSR	5.164674396
NVFEVGP GDS*PTFPR	5.164674396
RGNDPLTS*SPGR	5.164674396
RLS*YNTASNK	5.164674396
RTNS*TGGSSGSSVGGGSGK	5.164674396
SGAQASST*PLSPTR	5.164674396
SLSVS*SDFLGK	5.164674396
SSSSASPSS*PSSR	5.164674396
STSAPQMS*PGSSDNQSSSPQPAQQK	5.164674396
VLGS*EGEEDEALSPAK	5.164674396
TRHS*PT*PQQSNR	4.948612379
GLKEGMNPSYDEYADS*DEDQHDAYLER	4.87573437
IDS*PGFKPASQQK	4.87573437
IEVLPVDTGAGGYSGNS*GSPK	4.87573437
ISS*KSPGHMVILDQTK	4.87573437
KETQS*PEQVK	4.87573437
KTES*QKT*KSK	4.87573437
LFS*LSNPSLSTTNLSGPSR	4.87573437
LRLCDDGPQLPTS*PR	4.87573437
PSTQPRPDS*WGEDNWEGLTDSR	4.87573437
SKGHYEVT*GSDD*ETGK	4.87573437

SLS*ELESCLKLPAESNEK	4.87573437
SSHYGGS*LPNVNQIGSGLAEFQSPLHSPLDSSR	4.87573437
SSS*MAAGLER	4.87573437
TYS*LTPPAR	4.87573437
ASSHSSQTQGGGS*VTK	4.750842039
AKS*PT*PDGSER	4.708869205
GINGGPSRMS*PK	4.708869205
METEADAPS*PAPSLGER	4.708869205
VGG*DEEASGIPSR	4.708869205
VPLAPITDPQQLQLS*PLK	4.708869205
GGFDS*PFYR	4.263349417
NYQQNYQNSESGEKNES*ESAPEGQAQQR	4.263349417
QRS*PLLNPVPELSHASLIANQSPFR	4.263349417
AGS*ISTLDSLDFAR	4.190138967
HNS*TTSSTSSGGYR	4.190138967
KGS*ITEYTAEEK	4.190138967
KPIETGS*PKTK	4.190138967
RSS*LLNAK	4.190138967
LKDLFDYS*PPLHK	3.922285183
ADSGPTQPPLSLS*PAPETK	3.904932887
AES*PAEKVPEESVLPLVQK	3.904932887
AGKEPAKPS*PSR	3.904932887
AGLESGAEPGDGSDTT*KK	3.904932887
APVQPQGS*PAAAPGGTDEKPSGK	3.904932887
AQSS*PAAPASLSAPEPASQAR	3.904932887
ASS*LNVLNVGGK	3.904932887
ATT*PADGEEPAPAEALAAAR	3.904932887
DHS*PTPSVFNS*DEERYR	3.904932887
EAEEGPTGASESQDS*PR	3.904932887
EEDEPEERS*GDETPGSEVPGDKAAEEQGDDQDSEK	3.904932887
EQSEVSVS*PR	3.904932887
EYIPGQPPLSQSSDS*SPTR	3.904932887
GGTPAFLPSSLS*PQSSLPASR	3.904932887
GLLPGGTQVLDGTSGFS*PAPK	3.904932887
GRAS*PGGVSTSSSDGKAEK	3.904932887
HHNS*TAELQK	3.904932887
IAAPELHKGDS*DS*EEDEPTK	3.904932887
LDQPVSAPPS*PR	3.904932887
LQEEGGGS*DEEETGSPSEDGMQSAR	3.904932887
LSLTSDP EEGDPLALGPES*PGEPQPPQLK	3.904932887
LSVPT*SDEEDEVPAKPR	3.904932887
MSPNETLFLEST*NK	3.904932887
NLALDEAGQRS*TM	3.904932887
NQKPSQVNGAPGS*PTEPAGQK	3.904932887
PGPTPSGTNVGS*SGRSPSK	3.904932887
QAQAQES*EEEEESR	3.904932887
RAQS*TDSLGTSGSLQSK	3.904932887
RES*PS*PAPKPR	3.904932887
RGS*ASGSEPAGDSDR	3.904932887

RKGS*GS*EQEGEDEEGGER	3.904932887
RPLDS*PEAEELPAMK	3.904932887
RPS*PQPS*PR	3.904932887
RS*PS*PAPPPR	3.904932887
RS*PTGSTTSR	3.904932887
S*LDGAPIGVMDQSLMK	3.904932887
SAHTVEHGS*PR	3.904932887
SLPAPVAQRPDSPGGGLQAPGQK	3.904932887
SRTS*PVTR	3.904932887
SSGLST*PVPPSAGHLAHVR	3.904932887
SSGNSSSSGSGSGSTSAGSSS*PGAR	3.904932887
SSPNPFVGSPPKGLS*IQNGVK	3.904932887
SSS*ETILSSLAGSDIVK	3.904932887
STLQDS*DEYSNPAPLPLDQHSR	3.904932887
TGKEYIPGQPPLSQSSDS*SPTR	3.904932887
TS*PQVLGSILK	3.904932887
VASGSPGEGIS*PQSAQAPQAPGDHVVPLR	3.904932887

	50cGy
seq_top	GTestF=1.25
APEPLGPADQSELGPEQLEAEVGES*S*DEEPVESR	37.85609904
MLAES*DES*GDEESVSQTDKTELQNTLR	12.47860737
VADAKGDS*ES*EEDEDLEVPVPSR	11.54000068
TRHS*PT*PQQSNR	10.74792304
ASSHSSQTQGGGS*VTK	8.49678164
IVRGDQPAASGDS*DDDEPPPLPR	7.585575572
GAGATSGS*PPAGRN	7.385837878
TYS*LTPPAR	7.34968835
VLSKPPEGVVSEVEMLSS*QER	7.34968835
ALPAAAEDGS*PVFGEGPPSLK	7.218945525
AGGAS*PAASSTAQPPTQHR	6.2877469
ISYIPDEEVSSPS*PPQR	6.2877469
METEADAPS*PAPSLGER	6.2877469
QQPPLGPSSSLLS*LPGLK	6.2877469
SRAS*PATHR	6.2877469
VPLAPITDPQQQLQLS*PLK	6.2877469
RRS*PS*PPPTR	6.253157013
SGTAETEPVEQDSS*QPSLPLVR	6.10159956
SPMFPALGEASS*DDDLFQSAK	6.10159956
TSS*GTSLSAMHSSGSSGK	6.10159956
WLAES*PVGLPPEEEDKLTR	6.10159956
KVVDYSQFQES*DDADEDYGR	5.70972485
AVT*PVPTKTEEVSNLK	5.310580122
EGRGES*ENAGTNQETR	5.2060235
GSS*RQGS*PGSVSK	5.2060235
LGEEPEEEAQGPDAPAAAS*LPGS*PAPSQR	5.2060235
QIASQFPPPTPPAMESQPLKVPANVAPQS*PPAVK	5.2060235
RNS*LTGEEGLAR	5.2060235

SAS*ADNLTLP	5.2060235
SRS*DNALNLVTER	5.2060235
VPAS*PLPGLER	5.2060235
TNGHS*PSQPR	4.973813197
ARET*VENLPPLPLDPVLR	4.87573437
EGLGQQQS*LEQLEALVQTK	4.87573437
ESS*PLYS*PTFSDSTSAVK	4.87573437
GLKEGMNPSYDEYADS*DEDQHDAYLER	4.87573437
GLSAS*LPDLSENWIEVK	4.87573437
GMYDGPVFDLTTT*PKGCTPAGSAR	4.87573437
IPRPLS*LIGSTLR	4.87573437
ISKLEVTEIVKPS*PK	4.87573437
KVELS*ESEEDKGGK	4.87573437
NDELLSDLTRT*PPPPSSTFPK	4.87573437
S*LLESSLAGPGEDPLSADSLGKPTR	4.87573437
S*LSELESLKLPAESNEK	4.87573437
SLS*ELESKLPAESNEK	4.87573437
SSKAS*LGSLEGEAEAEASSPK	4.87573437
SSLS*GDEEDELFGATLK	4.87573437
TNS*MQQLEQWIK	4.87573437
RRS*QS*IEQESQEK	4.798360389
S*LDSDES*EDEEDDYQQK	4.707269063
NYQQNYQNSESGEKGES*ESAPEGQAQQR	4.641762401
SEVQQPVHPKPLSPDS*R	4.641762401
WLNSGRGDEASEEGQNGSS*PK	4.641762401
LAEDGE*EPEAVGQSR	4.395936385
RDS*PLQSGGQQNSQAGQR	4.395936385
SMS*AEDLLER	4.395936385
LQPSSS*PENSLDPFPPR	4.158049794
AERS*PNQGK	4.145857213
ARS*RT*PPSAPSQSR	4.145857213
AS*SHSSQTQGGGSVTK	4.145857213
ELVEPLT*PSGEAPNQALLR	4.145857213
FGLGS*PLPS*PR	4.145857213
GPPDFS*S*DEEREPT*PVLGSGAAAAGR	4.145857213
GPPS*PPAPVMHS*PSR	4.145857213
GRS*FAGNLNTYKR	4.145857213
HS*PTPQQSNR	4.145857213
KDS*QNSSQHSVSSHR	4.145857213
MQFS*FEGPEK	4.145857213
NHS*GS*RT*PPVALNSSR	4.145857213
RAGGGGLGAGS*PALSGGQGR	4.145857213
RLS*NVSLTG*VSTIR	4.145857213
RLS*YNTASNK	4.145857213
RTNS*TTGGSSGSSVGGGSGK	4.145857213
S*SPAAFINPPIGTVTPALK	4.145857213
SNSGLATYS*PPMGPVSER	4.145857213
SRNS*PLLER	4.145857213
SSSSSASPSS*PSSR	4.145857213

STQKS*PATAPK	4.145857213
STSAPQMS*PGSSDNQSSSPQPAQQK	4.145857213
TSTTGVATTQSPT*PR	4.145857213
VGGS*SVDLHR	4.145857213
VSPSKSPSLSPS*PPSPLEK	4.145857213
AS*LGSLEGEAEAEASSPK	3.883959757
AGS*ISTLDSLDFAR	3.860002193
EGMNPSYDEYADS*DEDQHDAYLER	3.860002193
KES*KEETPEVTK	3.860002193
QEASTGQSPEDHASLAPPLS*PDHSSLEAK	3.860002193